Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1264662, логином inf24f_139

Сводный итог по всем задачам в проверяющей системе

Run ID Time User name Problem Language Result Tests Score
9 0:10:47 inf24f_139 1 clang++ OK 28 100
65 1:12:15 inf24f_139 2 clang++ Partial solution 3 0
126 2:07:38 inf24f_139 3 g++ Partial solution 19 64
183 2:42:06 inf24f_139 4 g++ OK 21 100
242 2:59:20 inf24f_139 5 clang++ Partial solution 21 95
359 технических баллов
71 итоговый балл

Посылка по задаче 1

```
[1]  #include <bits/stdc++.h>
[2]  #define int long long
[3]  #pragma GCC optimize("Ofast")
[4]  #pragma GCC target("avx2,sse,sse2,sse3")
[5]
[6]  const int p = 129;
[7]  const int mod = 1e9 + 7;
[8]
[9]  struct Node
[10] {
[11]     int cnt;
[12] };
[13]
[14] struct SegmentTree
[15] {
[16]     int n;
[17]     std::vector<Node> t;
[18]
[19]     SegmentTree(std::vector<int> &a)
[20]     {
[21]         n = a.size();
[22]         t.resize(4 * n);
[23]         build(0, 0, n, a);
[24]     }
[25]
[26]     Node mer(Node a, Node b)
[27]     {
[28]         return {a.cnt + b.cnt};
[29]     }
[30]
[31]     void build(int i, int l, int r, std::vector<int> &a)
[32]     {
[33]         if (l == r - 1)
[34]         {
[35]             t[i].cnt = a[l];
[36]             std::cout << t[i].cnt << ' ' << a[l] << std::endl;
[37]             return;
[38]         }
[39]         int mid = (l + r) / 2;
[40]         build(2 * i + 1, l, mid, a);
[41]         build(2 * i + 2, mid, r, a);
[42]
[43]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[44]     }
[45]
[46]     void upd(int i, int l, int r, int pos, Node v)
[47]     {
[48]         if (l == r - 1)
[49]         {
[50]             t[i] = v;
[51]             return;
[52]         }
[53]         int mid = (l + r) / 2;
[54]         if (mid > pos)
[55]         {
[56]             upd(2 * i + 1, l, mid, pos, v);
[57]         }
[58]         else
[59]         {
[60]             upd(2 * i + 2, mid, r, pos, v);
[61]         }
[62]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[63]     }
[64]
[65]     Node query(int i, int l, int r, int ql, int qr)
[66]     {
[67]         if (l >= ql && r <= qr)
[68]         {
[69]             return t[i];
[70]         }
[71]         if (l >= qr || r <= ql)
[72]         {
[73]             return {0};
[74]         }
[75]         int mid = (l + r) / 2;
[76]         return mer(query(2 * i + 1, l, mid, ql, qr), query(2 * i + 2, mid, r, ql, qr));
[77]     }
[78] };
[79]
[80] int take_hash1(std::vector<int> &a)
[81] {
[82]     int hash = 0;
[83]     for (int i = 0; i < (int)a.size(); i++)
[84]     {
[85]         hash = (hash * p) % mod;
[86]         hash += a[i];
[87]         hash %= mod;
[88]     }
[89]     return hash;
[90] }
[91]
[92] std::vector<int> take_hash(std::vector<int> &a)
```

```
[93] {
[94]     std::vector<int> hash(a.size());
[95]     for (int i = 0; i < (int)a.size(); i++)
[96]     {
[97]         hash[i] = (hash[i - 1] * p) % mod;
[98]         hash[i] += a[i];
[99]         hash[i] %= mod;
[100]    }
[101]    return hash;
[102] }
[103]
[104] int fast_solve()
[105] {
[106] }
[107]
[108] int slow_solve()
[109] {
[110] }
[111]
[112] std::mt19937 rnd(121212);
[113] const int count_test = 1e5;
[114] int32_t main()
[115] {
[116]    std::ios::sync_with_stdio(false);
[117]    std::cin.tie(0);
[118]
[119]    std::vector<int> a;
[120]    a.push_back(0);
[121]    a.push_back(0);
[122]    a.push_back(1);
[123]    while (a.back() <= (1 << 25))
[124]    {
[125]        a.push_back(a.back() + a[a.size() - 2] + a[a.size() - 3]);
[126]    }
[127]
[128]    int n;
[129]    std::cin >> n;
[130]    int count = 0;
[131]    while (n--)
[132]    {
[133]        int num;
[134]        std::cin >> num;
[135]
[136]        std::vector<int> b(a.size(), 0);
[137]        int now = a.size() - 1;
[138]        int tmp = 0;
[139]        while (now >= 0)
[140]        {
[141]            if (num >= a[now])
[142]            {
[143]                b[now] = 1;
[144]                num -= a[now];
[145]                tmp++;
[146]            }
[147]            now--;
[148]        }
[149]        if (tmp % 2 != 0)
[150]        {
[151]            count++;
[152]        }
[153]    }
[154]    std::cout << count;
[155] }
```

Посылка по задаче 2

```
[1]  #include <bits/stdc++.h>
[2]  #define int long long
[3]  #pragma GCC optimize("Ofast")
[4]  #pragma GCC target("avx2,sse,sse2,sse3")
[5]
[6]  const int p = 129;
[7]  const int mod = 1e9 + 7;
[8]
[9]  struct Node
[10] {
[11]     int cnt;
[12] };
[13]
[14] struct SegmentTree
[15] {
[16]     int n;
[17]     std::vector<Node> t;
[18]
[19]     SegmentTree(std::vector<int> &a)
[20]     {
[21]         n = a.size();
[22]         t.resize(4 * n);
[23]         build(0, 0, n, a);
[24]     }
[25]
[26]     Node mer(Node a, Node b)
[27]     {
[28]         return {a.cnt + b.cnt};
[29]     }
[30]
[31]     void build(int i, int l, int r, std::vector<int> &a)
[32]     {
[33]         if (l == r - 1)
[34]         {
[35]             t[i].cnt = a[l];
[36]             std::cout << t[i].cnt << ' ' << a[l] << std::endl;
[37]             return;
[38]         }
[39]         int mid = (l + r) / 2;
[40]         build(2 * i + 1, l, mid, a);
[41]         build(2 * i + 2, mid, r, a);
[42]
[43]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[44]     }
[45]
[46]     void upd(int i, int l, int r, int pos, Node v)
[47]     {
[48]         if (l == r - 1)
[49]         {
[50]             t[i] = v;
[51]             return;
[52]         }
[53]         int mid = (l + r) / 2;
[54]         if (mid > pos)
[55]         {
[56]             upd(2 * i + 1, l, mid, pos, v);
[57]         }
[58]         else
[59]         {
[60]             upd(2 * i + 2, mid, r, pos, v);
[61]         }
[62]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[63]     }
[64]
[65]     Node query(int i, int l, int r, int ql, int qr)
[66]     {
[67]         if (l >= ql && r <= qr)
[68]         {
[69]             return t[i];
[70]         }
[71]         if (l >= qr || r <= ql)
[72]         {
[73]             return {0};
[74]         }
[75]         int mid = (l + r) / 2;
[76]         return mer(query(2 * i + 1, l, mid, ql, qr), query(2 * i + 2, mid, r, ql, qr));
[77]     }
[78] };
[79]
[80] int take_hash1(std::vector<int> &a)
[81] {
[82]     int hash = 0;
[83]     for (int i = 0; i < (int)a.size(); i++)
[84]     {
[85]         hash = (hash * p) % mod;
[86]         hash += a[i];
[87]         hash %= mod;
[88]     }
[89]     return hash;
[90] }
[91]
[92] std::vector<int> take_hash(std::vector<int> &a)
[93] {
```

```
    std::vector<int> hash(a.size());
    for (int i = 0; i < (int)a.size(); i++)
    {
        hash[i] = (hash[i - 1] * p) % mod;
        hash[i] += a[i];
        hash[i] %= mod;
    }
    return hash;
}

int fast_solve()
{
}

int slow_solve()
{
}

std::mt19937 rnd(121212);
const int count_test = 1e5;

std::map<char,long double> m;

void dfs(int start,std::string& a,long double now_sq){
    //std::cout<<now_sq<<std::endl;
    for(int i=start+1;i<std::min((int)a.size(),start+5);i++){
        if(a[i]=='Q'){
            dfs(i,a,now_sq/4);
        }else{
            m[a[i]]+=now_sq;
        }
    }
}

int32_t main()
{
    std::ios::sync_with_stdio(false);
    std::cin.tie(0);

    std::string a;
    std::cin>>a;

    if(a.size()==0){
        std::cout<<"0.0";
        return 0;
    }
    if(a.size()==1){
        std::cout<<a[0]<<std::endl;
        std::cout<<"1.0";
        return 0;
    }
    dfs(0,a,0.25);

    std::pair<long double,char> ans;

    for(std::pair<char,long double> i:m){
        if(i.second>=ans.first){
            ans.first=i.second;
            ans.second=i.first;
        }
    }

    std::cout<<ans.second<<std::endl;

    if(ans.first==1.0){
        std::cout<<"1.0";
        return 0;
    }

    long double now=0.5;
    std::string tmp="0.";
    int cnt=0;
    while(ans.first!=0 && cnt<=1000){
        cnt++;
        if(ans.first>=now){
            ans.first-=now;
            tmp.push_back('1');
        }else{
            tmp.push_back('0');
        }
        now/=2;
    }
    std::cout<<tmp;


}
```

Посылка по задаче 3

```
[1]  #include <bits/stdc++.h>
[2]  #define int long long
[3]
[4]
[5]
[6]  std::string to_digit(int num){
[7]
[8]      if(num==1){
[9]          return ">";
[10]     }else if(num==2){
[11]         return ">!";
[12]     }else if(num==3){
[13]         return ">!!";
[14]     }
[15]     else if(num==4){
[16]         return ">!!!";
[17]     }else if(num==5){
[18]         return ">?";
[19]     }
[20]      else if(num==6){
[21]         return "<";
[22]     }else if(num==7){
[23]         return "<!";
[24]     }
[25]     else if(num==8){
[26]         return "<!!";
[27]     }else if(num==9){
[28]         return "<!!!";
[29]     }
[30]     else if(num==10){
[31]         return "<?";
[32]     }else{
[33]         assert(0);
[34]     }
[35] }
[36]
[37] std::string conv_to_gnome(std::vector<int> num){
[38]     std::reverse(num.begin(),num.end());
[39]     std::string ans;
[40]     for(int i=0;i<num.size();i++){
[41]         if(num[i]==0){
[42]             int tmp_i=i;
[43]             while(tmp_i<num.size() && num[tmp_i]==0 ){
[44]                 tmp_i++;
[45]             }
[46]             assert(num[tmp_i]!=0);
[47]             num[tmp_i]--;
[48]             for(int j=tmp_i-1;j>i;j--){
[49]                 assert(num[j]==0);
[50]                 num[j]=9;
[51]             }
[52]             num[i]=10;
[53]         }
[54]         ans+=to_digit(num[i]);
[55]     }
[56]     return ans;
[57] }
[58]
[59] std::vector<int> to_vector (int num){
[60]     std::vector<int> ans;
[61]     while(num>0){
[62]         ans.push_back(num%10);
[63]         num/=10;
[64]     }
[65]     std::reverse(ans.begin(),ans.end());
[66]     return ans;
[67] }
[68]
[69] int to_int1(std::string a){
[70]     if(a==">"){
[71]         return 1;
[72]     }else if(a==">!"){
[73]         return 2;
[74]     }else if(a==">!!"){
[75]         return 3;
[76]     }else if(a==">!!!"){
[77]         return 4;
[78]     }else if(a==">?"){
[79]         return 5;
[80]     }else if(a=="<"){
[81]         return 6;
[82]     }else if(a=="<!"){
[83]         return 7;
[84]     }else if(a=="<!!"){
[85]         return 8;
[86]     }else if(a=="<!!!"){
[87]         return 9;
[88]     }else if(a=="<?"){
[89]         return 10;
[90]     }
[91] }
[92]
```

```
[93] std::vector<int> to_int(std::string a){
[94]     std::string now;
[95]     now+=a[0];
[96]     std::vector<int> ans;
[97]     int bit=0;
[98]     for(int i=1;i<a.size();i++){
[99]         if(a[i]=='>' || a[i]=='<'){
[100]            ans.push_back(to_int1(now));
[101]            now.clear();
[102]            now+=a[i];
[103]        }else{
[104]            now+=a[i];
[105]        }
[106]    }
[107]    ans.push_back(to_int1(now));
[108]    std::reverse(ans.begin(),ans.end());
[109]    return ans;
[110] }
[111]
[112] std::vector<int> to_normal_int(std::vector<int> a){
[113]    bool flg=false;
[114]    for(int i=a.size()-1;i>=0;i--){
[115]        if(a[i]>=10){
[116]            if(i==0){
[117]                flg=true;
[118]            }else{
[119]                a[i]-=10;
[120]                a[i-1]++;
[121]            }
[122]
[123]        }
[124]    }
[125]    if(flg){
[126]        std::vector<int> help;
[127]        help.push_back(1);
[128]        for(int i:a){
[129]            help.push_back(a[i]);
[130]        }
[131]        a=help;
[132]    }
[133]    return a;
[134] }
[135]
[136] bool more(std::vector<int>& a,std::vector<int>& b){
[137]    if(b.size()>a.size()){
[138]        return false;
[139]    }else if(a.size()>b.size()){
[140]        return true;
[141]    }
[142]    for(int i=0;i<a.size();i++){
[143]        if(a[i]!=b[i]){
[144]            return a[i]>b[i];
[145]        }
[146]    }
[147]    return false;;
[148] }
[149]
[150] bool equal(std::vector<int>& a,std::vector<int>& b){
[151]    if(b.size()>a.size()){
[152]        return false;
[153]    }else if(a.size()>b.size()){
[154]        return false;
[155]    }
[156]    for(int i=0;i<a.size();i++){
[157]        if(a[i]!=b[i]){
[158]            return false;
[159]        }
[160]    }
[161]    return true;;
[162] }
[163]
[164] int32_t main()
[165] {
[166]    std::ios::sync_with_stdio(false);
[167]    std::cin.tie(0);
[168]
[169]
[170]    int n;
[171]    std::cin>>n;
[172]    std::vector<std::vector<int>> tmp;
[173]    std::vector<int> max,min;
[174]    std::vector<int> index_max,index_min;
[175]    for(int i=0;i<n;i++){
[176]        std::string a;
[177]        std::cin>>a;
[178]
[179]        if(a=="()"){
[180]            tmp.push_back({0});
[181]        }else{
[182]            tmp.push_back(to_normal_int(to_int(a)));
[183]        }
[184]    }
```

```
[185]
[186]     for(int i=0;i<n;i++){
[187]         if(more(tmp[i],max) || max.size()==0){
[188]             index_max.clear();
[189]             index_max.push_back(i+1);
[190]             max=tmp[i];
[191]         }else if(equal(tmp[i],max)){
[192]             index_max.push_back(i);
[193]         }
[194]
[195]         if(more(min,tmp[i]) || min.size()==0){
[196]             index_min.clear();
[197]             index_min.push_back(i+1);
[198]             min=tmp[i];
[199]         }else if(equal(tmp[i],min)){
[200]             index_min.push_back(i);
[201]         }
[202]     }
[203]
[204]     std::pair<int,int> ans={1e9,1e9};
[205]     assert(index_max.size()!=0 && index_min.size()!=0);
[206]     for(int i:index_max){
[207]         for(int j:index_min){
[208]             if(ans.first+ans.second>i+j){
[209]                 ans.first=i;
[210]                 ans.second=j;
[211]             }
[212]         }
[213]     }
[214]     if(ans.second<ans.first){
[215]         ans={ans.second,ans.first};
[216]     }
[217]     std::cout<<conv_to_gnome(to_vector(ans.first))<<std::endl;
[218]     std::cout<<conv_to_gnome(to_vector(ans.second))<<std::endl;
[219]
[220] }
```

Посылка по задаче 4

```
#include <bits/stdc++.h>
#define int long long

int32_t main()
{
    std::ios::sync_with_stdio(false);
    std::cin.tie(0);


    int n,m;
    std::cin>>n>>m;
     int x1,y1,z1;
    std::cin>>x1>>y1>>z1;
    std::vector<std::vector<int>> a(n,std::vector<int>(m));
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            std::cin>>a[i][j];
        }
    }
    int r=n;
    int c=m;
    int dp[21][21][21];
    int prev_dp[21][21][21];
    for(int i=0;i<20;i++){
        for(int j=0;j<20;j++){
            for(int z=0;z<20;z++){
                dp[i][j][z]=-1e9;
                prev_dp[i][j][z]=-1e9;
            }
        }
    }

    prev_dp[x1][y1][z1]=a[x1][0];
    if(x1!=y1){
        prev_dp[x1][y1][z1]+=a[y1][0];
    }
    if(z1!=x1 && z1!=y1){
        prev_dp[x1][y1][z1]+=a[z1][0];
    }
    std::vector<int> dx={-1,0,1};
    //std::cout<<4343434434<<std::endl;
    for(int i=0;i<c-1;i++){
        for(int x=0;x<r;x++){
            for(int y=0;y<r;y++){
                for(int z=0;z<r;z++){
                    if(prev_dp[x][y][z]<0){
                        continue;
                    }

                    for(int first:dx){
                        for(int second:dx){
                            for(int third:dx){
                                if(first+x>=0 && first+x<r && second+y>=0 && second+y<r && third+z>=0 && third+z<r)
{

                                    int tmp1=a[first+x][i+1];
                                    int tmp2=0;
                                    if(first+x!=second+y){
                                        tmp2=a[second+y][i+1];
                                    }
                                    int tmp3=0;
                                    if(first+x!=third+z && second+y!=third+z){
                                        tmp3=a[third+z][i+1];
                                    }
                                    dp[first+x][second+y][third+z]=std::max(dp[first+x][second+y]
[third+z],prev_dp[x][y][z]+tmp1+tmp2+tmp3);
                                }
                            }
                        }
                    }

                }
            }
        }
        //std::cout<<i<<std::endl;


         for(int x=0;x<r;x++){
            for(int y=0;y<r;y++){
                for(int z=0;z<r;z++){

                    prev_dp[x][y][z]=dp[x][y][z];
                    if(i!=c-2){
                    dp[x][y][z]=-1e9;}
                }
            }
        }

    }
    int max=0;
    for(int x=0;x<r;x++){
        for(int y=0;y<r;y++){
            for(int z=0;z<r;z++){
                max=std::max(max,dp[x][y][z]);
```

```
[94]             }
[95]         }
[96]     }
         std::cout<<max;
     }
```

Посылка по задаче 5

```
[1]  #include <bits/stdc++.h>
[2]  #define int long long
[3]  #pragma GCC optimize("Ofast")
[4]  #pragma GCC target("avx2,sse,sse2,sse3")
[5]
[6]  const int p = 129;
[7]  const int mod = 1e9 + 7;
[8]
[9]  struct Node
[10] {
[11]     int cnt;
[12] };
[13]
[14] struct SegmentTree
[15] {
[16]     int n;
[17]     std::vector<Node> t;
[18]
[19]     SegmentTree(std::vector<int> &a)
[20]     {
[21]         n = a.size();
[22]         t.resize(4 * n);
[23]         build(0, 0, n, a);
[24]     }
[25]
[26]     Node mer(Node a, Node b)
[27]     {
[28]         return {a.cnt + b.cnt};
[29]     }
[30]
[31]     void build(int i, int l, int r, std::vector<int> &a)
[32]     {
[33]         if (l == r - 1)
[34]         {
[35]             t[i].cnt = a[l];
[36]             std::cout << t[i].cnt << ' ' << a[l] << std::endl;
[37]             return;
[38]         }
[39]         int mid = (l + r) / 2;
[40]         build(2 * i + 1, l, mid, a);
[41]         build(2 * i + 2, mid, r, a);
[42]
[43]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[44]     }
[45]
[46]     void upd(int i, int l, int r, int pos, Node v)
[47]     {
[48]         if (l == r - 1)
[49]         {
[50]             t[i] = v;
[51]             return;
[52]         }
[53]         int mid = (l + r) / 2;
[54]         if (mid > pos)
[55]         {
[56]             upd(2 * i + 1, l, mid, pos, v);
[57]         }
[58]         else
[59]         {
[60]             upd(2 * i + 2, mid, r, pos, v);
[61]         }
[62]         t[i] = mer(t[2 * i + 1], t[2 * i + 2]);
[63]     }
[64]
[65]     Node query(int i, int l, int r, int ql, int qr)
[66]     {
[67]         if (l >= ql && r <= qr)
[68]         {
[69]             return t[i];
[70]         }
[71]         if (l >= qr || r <= ql)
[72]         {
[73]             return {0};
[74]         }
[75]         int mid = (l + r) / 2;
[76]         return mer(query(2 * i + 1, l, mid, ql, qr), query(2 * i + 2, mid, r, ql, qr));
[77]     }
[78] };
[79]
[80] int take_hash1(std::vector<int> &a)
[81] {
[82]     int hash = 0;
[83]     for (int i = 0; i < (int)a.size(); i++)
[84]     {
[85]         hash = (hash * p) % mod;
[86]         hash += a[i];
[87]         hash %= mod;
[88]     }
[89]     return hash;
[90] }
[91]
[92] std::vector<int> take_hash(std::vector<int> &a)
[93] {
```

```
[94]      std::vector<int> hash(a.size());
[95]      for (int i = 0; i < (int)a.size(); i++)
[96]      {
[97]          hash[i] = (hash[i - 1] * p) % mod;
[98]          hash[i] += a[i];
[99]          hash[i] %= mod;
[100]     }
[101]     return hash;
[102] }
[103]
[104] int fast_solve()
[105] {
[106] }
[107]
[108] int slow_solve()
[109] {
[110] }
[111]
[112] std::mt19937 rnd(121212);
[113] const int count_test = 1e5;
[114]
[115]
[116] std::pair<int,int> Check(std::string& a,std::vector<int>& s,int len,int tmp){
[117]     std::vector<int> cnt(126-33+1,0);
[118]     int count=0;
[119]
[120]     for(int i=0;i<len;i++){
[121]         cnt[a[i]-33]++;
[122]         if(s[a[i]-33]==cnt[a[i]-33]){
[123]             count++;
[124]         }
[125]     }
[126]     if(count==tmp){
[127]         return {0,len-1};
[128]     }
[129]     for(int i=len;i<a.size();i++){
[130]         cnt[a[i-len]-33]--;
[131]          if(s[a[i-len]-33]==cnt[a[i-len]-33]+1){
[132]             count--;
[133]         }
[134]
[135]         cnt[a[i]-33]++;
[136]         if(s[a[i]-33]==cnt[a[i]-33]){
[137]             count++;
[138]         }
[139]         if(count==tmp){
[140]             return {i-len+1,i};
[141]         }
[142]     }
[143]     return {-1,-1};
[144] }
[145] int32_t main()
[146] {
[147]     std::ios::sync_with_stdio(false);
[148]     std::cin.tie(0);
[149]
[150]     // std::vector<int> a;
[151]     // a.push_back(0);
[152]     // a.push_back(0);
[153]     // a.push_back(1);
[154]     // while (a.back() <= (1 << 25))
[155]     // {
[156]     //     a.push_back(a.back() + a[a.size() - 2] + a[a.size() - 3]);
[157]     // }
[158]
[159]     // int n;
[160]     // std::cin >> n;
[161]     // int count = 0;
[162]     // while (n--)
[163]     // {
[164]     //     int num;
[165]     //     std::cin >> num;
[166]
[167]     //     std::vector<int> b(a.size(), 0);
[168]     //     int now = a.size() - 1;
[169]     //     int tmp = 0;
[170]     //     while (now >= 0)
[171]     //     {
[172]     //         if (num >= a[now])
[173]     //         {
[174]     //             b[now] = 1;
[175]     //             num -= a[now];
[176]     //             tmp++;
[177]     //         }
[178]     //         now--;
[179]     //     }
[180]     //     if (tmp % 2 != 0)
[181]     //     {
[182]     //         count++;
[183]     //     }
[184]     // }
```

```
[185]    // std::cout << count;
[186]
[187]    std::string a;
[188]    std::cin >> a;
[189]
[190]    std::string b;
[191]    std::cin >> b;
[192]    std::pair<int,int> ans={-1,-1};
[193]    if(a==b){
[194]        ans.first=0;
[195]        ans.second=b.size()-1;
[196]    }
[197]
[198]    int tmp1=0;
[199]
[200]    std::vector<int> used(126-33+1,0);
[201]
[202]    for(char c:b){
[203]        if(used[c-33]==0){
[204]            tmp1++;
[205]        }
[206]        used[c-33]++;
[207]    }
[208]
[209]
[210]    int l=0,r=a.size();
[211]    while(l+1<r){
[212]        int mid=(l+r)/2;
[213]
[214]
[215]        std::pair<int,int> tmp=Check(a,used,mid,tmp1);
[216]      //  std::cout<<mid<<' '<<tmp.first<<std::endl;
[217]
[218]
[219]        if(tmp.first!=-1){
[220]            ans=tmp;
[221]            r=mid;
[222]        }else{
[223]            l=mid;
[224]        }
[225]    }
[226]    if(ans.first==-1){
[227]        return 0;
[228]    }
[229]    for(int i=ans.first;i<=ans.second;i++){
[230]        std::cout<<a[i];
[231]    }
[232]
[233]
[234]
[235]
[236]
[237]
[238]
[239]
[240]
[241]
[242]
[243]
[244]
[245]    //  B
[246]    //  std::string a;
[247]    // std::cin>>a;
[248]
[249]    // if(a.size()==0){
[250]    //     std::cout<<"0.0";
[251]    //     return 0;
[252]    // }
[253]    // if(a.size()==1){
[254]    //     std::cout<<a[0]<<std::endl;
[255]    //     std::cout<<"1.0";
[256]    //     return 0;
[257]    // }
[258]    // dfs(0,a,0.25);
[259]
[260]    // std::pair<long double,char> ans;
[261]
[262]    // for(std::pair<char,long double> i:m){
[263]    //     if(i.second>=ans.first){
[264]    //         ans.first=i.second;
[265]    //         ans.second=i.first;
[266]    //     }
[267]    // }
[268]
[269]    // std::cout<<ans.second<<std::endl;
[270]
[271]    // if(ans.first==1.0){
[272]    //     std::cout<<"1.0";
[273]    //     return 0;
[274]    // }
[275]
[276]    // long double now=0.5;
[277]    // std::string tmp="0.";
```

```
[278]     // int cnt=0;
[279]     // while(ans.first!=0 && cnt<=1000){
[280]     //     cnt++;
[281]     //     if(ans.first>=now){
[282]     //         ans.first-=now;
[283]     //         tmp.push_back('1');
[284]     //     }else{
[285]     //         tmp.push_back('0');
[286]     //     }
[287]     //     now/=2;
[288]     // }
[289]     // std::cout<<tmp;
[290] }
```