

Contest Name : Day 1 總召好帥趣味賽

Starts at	Ends at	Contest Type
2021-01-27 19:00:00	2021-01-27 22:00:00	acm style (no partial/dashboard)

Description

注意事項

- 競賽時間共 180 分鐘。**每一隊每 5 秒只能上傳一次程式碼**。本次比賽將開放計分板，位於 Dashboard 的地方。若於題目上有疑慮，亦可以於以上 Q&A 的地方提問。
- 比賽會在最後 60 分鐘停止更新記分板。
- 每一題的題目有6個部分，分別是題目敘述（Description）、輸入格式（Input Format）、輸出格式（Output Format）、範例輸入（Sample Input）、範例輸出（Sample Output）、提示（Hints），請仔細閱讀後再進行作答。
- 本次競賽使用線上評測系統 TIOJ Infor Online Judge IOICamp edition（即本站）。參賽者需將程式碼上傳至線上評測系統，才予以計分。成績以評測系統上的分數為準。
（若發現測資有誤等情形，仍有可能在比賽中或結束後重新測試，且以重新測試後之分數作為最終分數。）
- 本次比賽，可以用 C++14、C++17、C11 等語言作答。
- 競賽的開始、結束時間以線上評測系統為準，只有在競賽期間上傳的程式碼才會計分。（以上傳程式碼的時間為準。若在競賽期間內上傳，但評測完畢時競賽已結束，仍然會計分。）
- 若題目內容有修正，將會發佈公告，並以線上評測系統的題目內容為主。
- 測驗期間，學員應隨時備份資料，以防資料流失。
- 測驗中只允許開啟撰寫程式必需的編輯器、輔助程式（限記事本、小算盤、小畫家），或使用 Discord 與隊上成員交流，不得開啟其他無關的應用程式，也不得連結本站以外的其他網站（下列的 STL 介紹不在此限），違者視同作弊，但我也不知道作弊會怎樣。
- 測驗中可參考 **C++ 標準模版庫（STL）** 介紹。本介紹並未包含 C++11 以後新增的功能。
- 測驗期間，請勿交談或與其他隊的學員有任何交流，違者視同作弊，但我還是不知道作弊會怎樣。
- 測驗期間，禁止攜帶或使用任何電子設備以及使用無線網路，違者視同作弊，所以說我還是不知道作弊會怎樣。
- 測驗期間，禁止以任何方式攻擊學校網路或評測系統，違者視同作弊，我...算了。
- 由於線上評測系統採自動化評分，**請參賽者務必看清題目敘述（包含範例輸入、輸出與提示），並嚴格遵守各題的輸入輸出格式**。若格式錯誤，將可能被判斷為 **WA（答案錯誤）**，並獲得 0 分。
- 若對評測得到的結果不甚了解，可參考**傳送結果介紹**。
- 競賽禁止使用自備的鍵盤、滑鼠等配件，請使用競賽場所所提供的設備。
- 總召好帥。
- 上傳前如果沒有大喊一聲「總召好帥」讓全體參加人員聽到，該筆 submission 可能會被評為 **WA（答案錯誤）**。

glhf OwO/

Tasks

pA 1. 警察辦公（除蟲題）

pB 2. 警察辦公（除蟲題）- EX

pC 3. Bozo Sort（除蟲題）

pD 4. Bozo Sort（除蟲題）- EX

pE 5. 特別週特別喜歡吃紅蘿蔔（駭客題）

pF 6. 東海帝皇 DNA 序列分析（駭客題）

pG 7. 最近點對 - Easy（駭客題）

pH 8. 最近點對 - Hard（駭客題）

pI 9. 璃奈的煩惱（駭客題）

pJ 10. 有錢人小 Y 過馬路（通靈題）

pK 11. 吐鈔機（通靈題）

pL 12. Deja Vu?（通靈題）

pM 13. Nekomancer（通靈題）

pN 14. 順基因方程式 - Easy（通靈題）

pO 15. 順基因方程式 - Mid（通靈題）

pP 16. 順基因方程式 - Hard（通靈題）

1. 警察辦公（除蟲題）

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

80.0% (4/5)

Tags

Description

小Y是一個警察，他住在一張形如 N 點 M 邊的帶權有向圖的城鎮內，其中，他家的地點是 H ，他上班的警察局是 P ，不過為了方便起見， H 的編號固定是 1， P 的編號固定是 N 。

小Y想要知道從 H 到 P 的最短路徑長是多少，好估計他平常得花多少時間上班，但由於小Y是一個優秀的警察，他常常會在若干天早上接到任務，這讓他從家裡出發後，必須先到達地點 u 再去地點 v ，最後才能去警察局上班。因此，對於每個任務，他也在想當下知道他能完成這些任務的最短路徑長是多少。

你可以寫一支程式，讀入城鎮的資訊後，幫助他回答路徑長嗎？

聰明的小P一看到題目就馬上寫了以下的 code 出來，但他吃了滿滿的 Wrong Answer，請給出與以下的程式碼 edit distance 不超過 5，並獲得 Accepted 的程式碼。

```
#include<bits/stdc++.h>
using namespace std;

const int INF = 1e9;
int n, dis[505][505];

void dp() {
    // floyd-warshall algorithm
    for (int i = 1; i <= n; ++i)
        for (int k = 1; k <= n; ++k)
            for (int j = 1; j <= n; ++j)
                dis[i][j] = min(dis[i][j], dis[i][k] + dis[k][j]);
}

int main() {
    int m, h, p, a, b, d;
    // input N M
    scanf("%d%d", &n, &m);
    h = 1, p = n;
    // initialize each weight of edges to INF
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= n; ++j)
            dis[i][j] = INF;
    // i -> i is zero
    for (int i = 1; i <= n; ++i)
        dis[i][i] = 0;
    // input edges
    while (m--) {
        scanf("%d%d%d", &a, &b, &d);
        // don't forget the multiple edges!
        dis[a][b] = min(dis[a][b], d);
    }
    // run floyd-warshall
    dp();
    // input h and p, actually they remain the same
    scanf("%d%d", &h, &p);
    // print the distance from h to p
    printf("%d\n", dis[h][p]);
    while (scanf("%d%d", &a, &b) != EOF) {
        // output the answer for each query
        if (dis[h][a] == INF || dis[a][b] == INF || dis[b][p] == INF)
            printf("-1\n");
        else
            printf("%d\n", dis[h][a] + dis[a][b] + dis[b][p]);
    }
}
```

Input Format

輸入的第一行有兩個正整數 N, M ，代表城鎮的點數和邊數。

接下來 M 行，每行三個正整數 a, b, d ，代表有一條有向邊可以從 a 走到 b ，且花費是 d 。

緊接著一行兩個正整數 H, P ，代表小Y家地點的編號和他上班的警察局的編號。

接下來若干行，每行兩個正整數 u, v ，代表小Y這次接到的任務，由於他不知道他會接到多少任務，所以請你讀到 EOF 結束。

注意輸入可能有自環，也有可能有多重邊。

- $1 \leq N \leq 500$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq a, b, u, v \leq N$
- $1 \leq d \leq 10^6$
- $H = 1$
- $P = N$
- $u \neq v$
- $\{u, v\} \cap \{H, P\} = \emptyset$
- 保證 H 可以走到 P

Output Format

首行輸出一個整數， H 到 P 的最短路徑長是多少。

接下來若干行，依序代表每個任務所需的最短路徑長是多少，若不存在完成任務的路徑，請輸出 -1 。

Sample Input

copy

```
4 4
1 2 3
2 3 3
3 4 3
1 4 3
1 4
2 3
```

Sample Output

copy

```
3
9
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				^
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	

2. 警察辦公 (除蟲題) - EX


Submit

Status

Ranklist

Back to Problems List

TopCoder



林敬珣、官毓韋、
王政祺
Shao, Dou Shao

User's AC Ratio

100.0% (4/4)

Submission's AC Ratio

62.5% (5/8)

Tags

Description

小Y是一個警察，他住在一張形如 N 點 M 邊的帶權有向圖的城鎮內，其中，他家的地點是 H ，他上班的警察局是 P ，不過為了方便起見， H 的編號固定是 1， P 的編號固定是 N 。

小Y想要知道從 H 到 P 的最短路徑長是多少，好估計他平常得花多少時間上班，但由於小Y是一個優秀的警察，他常常會在若干天早上接到任務，這讓他從家裡出發後，必須先到達地點 u 再去地點 v ，最後才能去警察局上班。因此，對於每個任務，他也在想當下知道他能完成這些任務的最短路徑長是多少。

你可以寫一支程式，讀入城鎮的資訊後，幫助他回答路徑長嗎？

聰明的小P一看到題目就馬上寫了以下的 code 出來，但他吃了滿滿的 Wrong Answer，請給出與以下的程式碼 edit distance 不超過 2，並獲得 Accepted 的程式碼。

```
#include<bits/stdc++.h>
using namespace std;

const int INF = 1e9;
int n, dis[505][505];

void dp() {
    // floyd-warshall algorithm
    for (int i = 1; i <= n; ++i)
        for (int k = 1; k <= n; ++k)
            for (int j = 1; j <= n; ++j)
                dis[i][j] = min(dis[i][j], dis[i][k] + dis[k][j]);
}

int main() {
    int m, h, p, a, b, d;
    // input N M
    scanf("%d%d", &n, &m);
    h = 1, p = n;
    // initialize each weight of edges to INF
    for (int i = 1; i <= n; ++i)
        for (int j = 1; j <= n; ++j)
            dis[i][j] = INF;
    // i -> i is zero
    for (int i = 1; i <= n; ++i)
        dis[i][i] = 0;
    // input edges
    while (m--) {
        scanf("%d%d%d", &a, &b, &d);
        // don't forget the multiple edges!
        dis[a][b] = min(dis[a][b], d);
    }
    // run floyd-warshall
    dp();
    // input h and p, actually they remain the same
    scanf("%d%d", &h, &p);
    // print the distance from h to p
    printf("%d\n", dis[h][p]);
    while (scanf("%d%d", &a, &b) != EOF) {
        // output the answer for each query
        if (dis[h][a] == INF || dis[a][b] == INF || dis[b][p] == INF)
            printf("-1\n");
        else
            printf("%d\n", dis[h][a] + dis[a][b] + dis[b][p]);
    }
}
```

Input Format

輸入的第一行有兩個正整數 N, M ，代表城鎮的點數和邊數。

接下來 M 行，每行三個正整數 a, b, d ，代表有一條有向邊可以從 a 走到 b ，且花費是 d 。

緊接著一行兩個正整數 H, P ，代表小Y家地點的編號和他上班的警察局的編號。

接下來若干行，每行兩個正整數 u, v ，代表小Y這次接到的任務，由於他不知道他會接到多少任務，所以請你讀到 EOF 結束。

注意輸入可能有自環，也有可能有多重邊。

- $1 \leq N \leq 500$
- $1 \leq M \leq 2 \times 10^5$
- $1 \leq a, b, u, v \leq N$
- $1 \leq d \leq 10^6$
- $H = 1$
- $P = N$
- $u \neq v$
- $\{u, v\} \cap \{H, P\} = \emptyset$
- 保證 H 可以走到 P

Output Format

首行輸出一個整數， H 到 P 的最短路徑長是多少。

接下來若干行，依序代表每個任務所需的最短路徑長是多少，若不存在完成任務的路徑，請輸出 -1 。

Sample Input

copy

```
4 4
1 2 3
2 3 3
3 4 3
1 4 3
1 4
2 3
```

Sample Output

copy

```
3
9
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	
1	1000	262144	65536	
2	1000	262144	65536	
3	1000	262144	65536	
4	1000	262144	65536	
5	1000	262144	65536	
6	1000	262144	65536	

3. Bozo Sort (除蟲題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



王禴立、陳泰穎、楊翔宇

User's AC Ratio

75.0% (3/4)

Submission's AC Ratio

42.9% (3/7)

Tags

Description

有人聽過Bozo Sort嗎？這是一個經典的排序演算法，而且有最佳時間複雜度 $O(N)$ ！然而平均複雜度是 $O(N!)$ 實務上常常會TLE。以下為一個Bozo Sort演算法但是會拿到TLE。請給出 edit distance 與下列的 code 不超過 5 的解。

```
#include<iostream>
#include<algorithm>
#include<random>

using namespace std;

int a[1005];

int main(){
    int n; cin>>n;
    for(int i=0;i<n;i++) cin>>a[i];
    mt19937 gen(880301);
    uniform_int_distribution<int> dis(0,n-1);
    int i=0,j=0,c=0;
    while(!is_sorted(a,a+n)){
        c++;
        if((i=dis(gen)),(j=dis(gen));a[i]!=a[j])
            swap(a[i],a[j]);
    }
    for(int i=0;i<n;i++) cout<<a[i]<<" \n"[i+1==n];
    return 0;
}
```

Input Format

輸入第一行有一個正整數 N 代表數列長度。
第二行有 N 個正整數 A_i 為數列內容。

- $1 \leq N \leq 1000$
- $1 \leq A_i \leq 10000000$

Output Format

輸出一行 N 個數字為數列排序後的結果。

Sample Input

copy

```
10
5 7 4 2 6 5 7 4 3 5
```

Sample Output

copy

```
2 3 4 4 5 5 5 6 7 7
```

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	65536	65536	
1	1000	65536	65536	
2	1000	65536	65536	
3	1000	65536	65536	
4	1000	65536	65536	
5	1000	65536	65536	
6	1000	65536	65536	
7	1000	65536	65536	
8	1000	65536	65536	
9	1000	65536	65536	

✔ 4. Bozo Sort (除蟲題) - EX

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

80.0% (4/5)

Tags

Description

有人聽過Bozo Sort嗎？這是一個經典的排序演算法，而且有最佳時間複雜度 $O(N)$ ！然而平均複雜度是 $O(N!)$ 實務上常常會 TLE 。以下為一個Bozo Sort演算法但是會拿到 TLE 。
請給出 edit distance 與下列的 code 不超過 3 的解。

```
#include<iostream>
#include<algorithm>
#include<random>

using namespace std;

int a[1005];

int main(){
    int n; cin>>n;
    for(int i=0;i<n;i++) cin>>a[i];
    mt19937 gen(880301);
    uniform_int_distribution<int> dis(0,n-1);
    int i=0,j=0,c=0;
    while(!is_sorted(a,a+n)){
        c++;
        if((i=dis(gen)),(j=dis(gen));a[i]!=a[j])
            swap(a[i],a[j]);
    }
    for(int i=0;i<n;i++) cout<<a[i]<<" \n"[i+1==n];
    return 0;
}
```

Input Format

輸入第一行有一個正整數 N 代表數列長度。
第二行有 N 個正整數 A_i 為數列內容。

- $1 \leq N \leq 1000$
- $1 \leq A_i \leq 10000000$

Output Format

輸出一行 N 個數字為數列排序後的結果。

Sample Input

copy

10
5 7 4 2 6 5 7 4 3 5

Sample Output

copy

2 3 4 4 5 5 5 6 7 7

Hints

Problem Source

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⤴
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	65536	65536	
1	1000	65536	65536	
2	1000	65536	65536	
3	1000	65536	65536	
4	1000	65536	65536	
5	1000	65536	65536	
6	1000	65536	65536	
7	1000	65536	65536	
8	1000	65536	65536	
9	1000	65536	65536	

5. 特別週特別喜歡吃紅蘿蔔 (駭客題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (2/2)

Submission's AC Ratio

100.0% (2/2)

Tags

Description

Original Description

這是擁有繼承自異世界的光輝名號及競速能力的「賽馬娘」自古以來就與人類共存的世界的故事。賽馬娘·特別週從鄉下轉學到位在都市的特雷森學園，她在與同班同學們一同切磋琢磨的同時，賭上「日本第一賽馬娘」的稱號，以在「Twinkle Series」當中獲勝為目標。

今天特雷森學園的食堂發起了大胃王比賽，提供 N 根紅蘿蔔免費給賽馬娘們吃，第 i 根紅蘿蔔的重量是 w_i 公斤，因為特別週特別喜歡吃紅蘿蔔，一眼就能看出第 i 根紅蘿蔔的美味度是 v_i ，但是她的肚子只放得下 M 公斤的紅蘿蔔，請問特別週該怎麼選擇要吃的紅蘿蔔，才能使吃下的紅蘿蔔美味度總和最高？

Original Input Format

第一行包含兩個正整數 M 跟 N ，分別代表特別週肚子的容量和紅蘿蔔根數，之後的第 i 行包含兩個正整數 v_i 跟 w_i ，分別代表紅蘿蔔美味度和紅蘿蔔重量。

- $1 \leq M \leq 5 \times 10^6$
- $1 \leq N \leq 1000$
- $1 \leq v_i \leq 10^5$
- $1 \leq w_i \leq 10^5$

Original Output Format

請輸出一個數字，代表特別週能吃下的紅蘿蔔的美味度的和的最大值。

Original Sample Input

Sample Input 1:
500 10
45 4
94 63
43 9
74 86
12 4
69 74
53 47
24 98
30 94
32 65

Sample Input 2:
1000 20
113 94
129 94
118 97
119 92
127 100
119 99
127 96
122 98
111 99
109 100
115 98
119 96
129 96
127 91
103 97
115 93
115 90
130 96
109 99
130 95

Original Sample Output

Sample Output 1:
452

Sample Output 2:
1259

Original Limits

- Time Limit: 1 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include<bits/stdc++.h>
using namespace std;
int m, n;
vector<pair<int, int>> a;
vector<int> sw, sv, minw;
struct node
{
    int w0; // 剩下的容量
    int v0; // 已經拿的美味度和
    int k;
    int lb; // lower bound
    int ub; // upper bound
    bool operator < (const node & r) const
    {
        return ub < r.ub;
    }
    void cal()
    {
        int l = k, r = n;
        while (l <= r) {
            int mid = (l + r) / 2;
            if (sw[mid] - sw[k] <= w0)
                l = mid + 1;
            else
                r = mid - 1;
        }
        ub = lb = v0 + sv[r] - sv[k];
        int w = w0 - (sw[r] - sw[k]);
        if (r + 1 <= n) {
            int t = min(w, a[r].second);
            ub += t * 1. * a[r].first / a[r].second;
        }
    }
};
int ans;
void dfs(node u)
{
    u.cal();
    ans = max(ans, u.lb);
    if (u.ub <= ans) // 若之後可能的最佳解不比已知的最佳解好就剪枝
        return;
    if (u.k == n)
        return;
    if (u.w0 < minw[u.k])
        return;
    int k = u.k++;
    node u0 = u;
    if (a[k].second <= u.w0) {
        u.w0 -= a[k].second;
        u.v0 += a[k].first;
        node u1 = u;
        dfs(u1), dfs(u0);
    } else {
        dfs(u0);
    }
}
int main()
{
    ios::sync_with_stdio(0);
    cin.tie(0);
    cin >> m >> n;
    a.resize(n);
    for (auto & p : a)
        cin >> p.first >> p.second;
    // 照 CP 值由大到小排序，一定很對！
    sort(begin(a), end(a), [](pair<int, int> i, pair<int, int> j) {return i.first * 1ll * j.second > j.first * 1ll * i.second;});
    minw = sv = sw = vector<int>(n + 1);
    for (int i = 1; i <= n; ++i) {
        sv[i] = a[i - 1].first + sv[i - 1];
        sw[i] = a[i - 1].second + sw[i - 1];
    }
    minw[n - 1] = a[n - 1].second;
    for (int i = n - 2; i >= 0; --i)
        minw[i] = min(a[i].second, minw[i + 1]);
    ans = 0;
    for (int i = 0, w = m; i < n && w > 0; ++i)
        if (a[i].second <= w) {
            w -= a[i].second;
            ans += a[i].first;
        }
    node u;
    u.w0 = m;
    u.k = u.v0 = 0;
    dfs(u);
    cout << ans << '\n';
}
```

Judge Method

極責是臺北帝雉大學四年級學生，雖然就讀情報科學工程學系，但既不喜歡情報，也不喜歡科學，更是討厭工程。住在「三疊宿舍」的二樓，廁所對面的房間。喜歡去帝雉大學附近森林中出現的貓拉麵麵攤吃拉麵。以「過著有如玫瑰色的校園生活」為期許加入了社團「弟子招募」，跟「師父」學習演算法。

有天極責在研究上面的問題時，師父交給他上面那份具有台灣價值的暴搜剪枝程式碼，極責隨便生了一些測試資料後發現它跑得飛快，甚至比極責自己寫的 DP 解法還快。極責覺得很生氣，怎麼可以讓這種不正義的程式 AC，請幫極責生一份測試資料上面的程式 TLE。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <cstdio>
int main() {
    printf("1 1\n1 1\n");
}
```

Input Format

Output Format

Sample Input

copy

Sample Output

copy

Hints

Problem Source

Subtasks

No.

Testdata Range

Score

Testdata and Limits

No.

Time Limit (ms)

Memory Limit (KiB)

Output Limit (KiB)

Subtasks

0

1000

65536

65536

6. 東海帝皇 DNA 序列分析 (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



盧沛宏、林立上、林煜傑

User's AC Ratio

100.0% (5/5)

Submission's AC Ratio

83.3% (5/6)

Tags

Description

Original Description

這是擁有繼承自異世界的光輝名號及競速能力的「賽馬娘」自古以來就與人類共存的世界的故事。憧憬著魯鐸象徵、目標成為無敗三冠賽馬娘的 *東海帝皇*，以及賭上名門榮耀、向天皇賞連霸發起挑戰的 *月白麥昆*。既是競爭對手又是朋友的兩人，賭上各自的夢想在「Twinkle Series」當中不斷奔跑。

東海帝皇 想知道自己的 DNA 跟 *皇帝* 的 DNA 有多相似，請幫她寫程式計算吧！

具體來說，給你兩個字串 s, t ，請輸出它們的「最長共同子序列」(LCS) 的長度與數量。

一個字串的子序列，是由字串中若干個字元(不一定連續)以相同的順序所組成的字串，例如：YP、MY、MIT 都是 MYLITTLEPONY 的子序列。兩個字串的最長共同子序列 (LCS) 即為兩個字串的共同子序列中長度最大者。

令 k 為 LCS 的長度，我們說 $a = \{a_1, a_2, \dots, a_k\}, b = \{b_1, b_2, \dots, b_k\}$ 表示一組 LCS，若 a, b 滿足 a 是 $\{1, 2, \dots, |s|\}$ 的子序列且 b 是 $\{1, 2, \dots, |t|\}$ 的子序列且 $s_{a_i} = t_{b_i} \forall i \in \{1, 2, \dots, k\}$ 。LCS 的數量即為合法的 (a, b) 數量。

Original Input Format

- 第一行包含一個字串 s 。
- 第二行包含一個字串 t 。
- 兩個字串都只由大寫英文字母組成。
- $1 \leq |s|, |t| \leq 100$

Original Output Format

請輸出一行為兩個由空格隔開的數字，第一個是 LCS 的長度，第二個是 LCS 的數量。

Original Sample Input

```
Sample Input 1:
MYLITTLEPONY
FRIENDSHIPISMAGIC

Sample Input 2:
AACTTTAA
GAAATTA
```

Original Sample Output

```
Sample Output 1:
3 2

Sample Output 2:
5 18
```

Original Limits

- Time Limit: 1 second
- Memory Limit: 256 MB

Program To Be Hacked

```
#include<bits/stdc++.h>
using namespace std;
#define int long long
const int N = 105;
string a, b;
int n, m;
int dp[N][N], ways[N][N];
int32_t main()
{
    ios::sync_with_stdio(0);
    cin.tie(0);
    cin >> a >> b;
    n = a.size(), m = b.size();
    for (int i = n - 1; i >= 0; --i)
        for (int j = m - 1; j >= 0; --j) {
            if (a[i] == b[j]) {
                dp[i][j] = 1 + dp[i + 1][j + 1];
                ways[i][j] = dp[i][j] == 1 ? 1 : ways[i + 1][j + 1];
            }
            if (dp[i][j + 1] >= dp[i][j]) {
                if (dp[i][j + 1] > dp[i][j])
                    ways[i][j] = 0;
                dp[i][j] = dp[i][j + 1];
                ways[i][j] += ways[i][j + 1];
            }
            if (dp[i + 1][j] >= dp[i][j]) {
                if (dp[i + 1][j] > dp[i][j])
                    ways[i][j] = 0;
                dp[i][j] = dp[i + 1][j];
                ways[i][j] += ways[i + 1][j];
            }
            if (dp[i + 1][j + 1] == dp[i][j]) {
                ways[i][j] -= ways[i + 1][j + 1];
            }
        }
    cout << dp[0][0] << ' ' << ways[0][0] << '\n';
}
```

Judge Method

極貴是臺北帝雉大學四年級學生，雖然就讀情報科學工程學系，但既不喜歡情報，也不喜歡科學，更是討厭工程。住在「三疊宿舍」的二樓，廁所對面的房間。喜歡去帝雉大學附近森林中出現的貓拉麵麵攤吃拉麵。以「過著有如玫瑰色的校園生活」為期許加入了社團「弟子招募」，跟「師父」學習演算法。

有天極貴在研究上面的問題時，師父交給他上面那份 DP 程式碼，雖然極貴一眼就發現它有重大的缺失，但極貴隨便生了一些測試資料後發現它都能正確回答。極貴覺得很生氣，怎麼可以讓這種不正義的程式 **AC**，請幫極貴生一份測資讓上面的程式 **WA**。

Sample Code

以下程式碼能產生本題合法但一定不會讓你得到 AC 的測試資料。

```
#include <cstdio>
int main() {
    printf("A\nA\n");
}
```

Input Format

Sample Input

copy

Hints

Output Format

Sample Output

copy

Problem Source

Subtasks

No.	Testdata Range		Score	
Testdata and Limits				
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	65536	65536	

7. 最近點對 - Easy (駭客題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



王榆立、陳泰穎、楊翔宇

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

60.0% (3/5)

Tags

Description

首先先來介紹一個經典問題：

給出二維平面上 N 個點，定義兩個點 $a_1 = (x_1, y_1), a_2 = (x_2, y_2)$ 的距離為 $Dis(a_1, a_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ，請找出 N 個點中最近的點對距離。

你說用分治法可以做嗎？小孩子才用分治！你從路邊撿來的報紙上面這樣寫，接著報紙上介紹了一個最近點對的做法：

1. 首先隨機選擇一個角度 θ ，然後將所有點以原點旋轉 θ 度。
2. 將這 N 個點依照 x 軸排序， x 軸相同的話照 y 軸排序，排序後我們稱這些點依序為 a_1, a_2, \dots, a_N 。
3. 決定一個神祕數字 k ，然後對於每個點 a_i 計算出該點與 $a_{i+1}, a_{i+2}, \dots, a_{i+k}$ 這些點中的最近點對距離 b_i ，接著我們便說最小的 b_i 就是我們找到的 N 個點中的最近點對。

正式來說，報紙上的做法說要計算：

$$\min_{i=1}^N \left(\min_{j=i+1}^{\min(j+k,N)} Dis(a_i, a_j) \right)$$

你想了一陣子之後覺得，小孩子才看報紙，上面的一定是錯的！因此請你構造一份測資讓底下的程式碼出錯。

報紙上還附了一份實作的程式碼，最後會輸出最近點對距離的平方。

附帶一提，底下的程式碼 **正確的** 實作了上面提到的做法，並不需要尋找實作上的錯誤。另外仔細觀察後便會發現，並不存在一個測資 **一定** 會讓報紙上的做法出錯，但可以讓報紙上的做法 **高機率** 的出錯，本題會有 10 筆測資，只要在 10 筆測資中都讓底下的程式碼出錯即可 **AC**，本題不會無故 Rejudge。程式碼中 `chrono::steady_clock::now().time_since_epoch().count()` 是一個精度為 10^{-9} 秒的時間函數，在程式中作為 random seed，確保每次 judge 都會旋轉不同的角度。

- Easy version:
 - 一開始不選轉點，即 $\theta = 0$ 。
 - $k = 200$

Original Sample Input

```
2
0 0
0 2
```

Original Sample Output

```
4
```

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;
#define maxn 200005

int n;
pair<int, int> p[maxn];
long double angle, cos_v, sin_v;

bool cmp(pair<int, int> a, pair<int, int> b) {
    auto v1 = make_pair(a.first * cos_v - a.second * sin_v, a.first * sin_v + a.second * cos_v);
    auto v2 = make_pair(b.first * cos_v - b.second * sin_v, b.first * sin_v + b.second * cos_v);
    return v1 < v2;
}

int main() {
    cin.tie(0), cout.sync_with_stdio(0);
    mt19937 rng(chrono::steady_clock::now().time_since_epoch().count()); // random number generator
    cin >> n;
    for (int i = 1; i <= n; ++ i)
        cin >> p[i].first >> p[i].second;

    long long ans = 6e18, k = 200;
    angle = rng(), angle = 0;
    cos_v = cos(angle);
    sin_v = sin(angle);
    sort(p + 1, p + 1 + n, cmp);
    for (int i = 1; i <= n; ++ i) { // Sweep backward k points
        for (int j = i + 1; j <= i + k && j <= n; ++ j) {
            long long v1 = (p[i].first - p[j].first);
            long long v2 = (p[i].second - p[j].second);
            long long dis = v1 * v1 + v2 * v2;
            ans = min(ans, dis);
        }
    }
    cout << ans << "\n";
    return 0;
}
```

Input Format

本題沒有輸入。

Output Format

第一行請輸出一個數字 N ，代表點數。
接下來一共有 N 行，請輸出兩個數字 x_i, y_i 代表第 i 個點的座標。

- $2 \leq N \leq 2 \times 10^5$
- $|x_i|, |y_i| \leq 10^9$

Sample Input

copy

Sample Output

copy

Hints

Sample Code

以下是一個格式正確但不會 **AC** 的範例程式

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    puts("2");
    puts("0 0");
    puts("0 1");
    return 0;
}
```

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0~9	100

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	5000	262144	65536	1	
1	5000	262144	65536	1	
2	5000	262144	65536	1	
3	5000	262144	65536	1	
4	5000	262144	65536	1	
5	5000	262144	65536	1	
6	5000	262144	65536	1	
7	5000	262144	65536	1	
8	5000	262144	65536	1	
9	5000	262144	65536	1	

8. 最近點對 - Hard (駭客題)

Submit

Status

Ranklist

Back to Problems List

TopCoder

User's AC Ratio

Submission's AC Ratio

Tags

NaN% (0/0)

NaN% (0/0)

Description

首先先來介紹一個經典問題：

給出二維平面上 N 個點，定義兩個點 $a_1 = (x_1, y_1), a_2 = (x_2, y_2)$ 的距離為 $Dis(a_1, a_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ，請找出 N 個點中最近的點對距離。

你說用分治法可以做嗎？小孩子才用分治！你從路邊撿來的報紙上面這樣寫，接著報紙上介紹了一個最近點隊的做法：

- 首先隨機選擇一個角度 θ ，然後將所有點以原點旋轉 θ 度。
- 將這 N 個點依照 x 軸排序， x 軸相同的話照 y 軸排序，排序後我們稱這些點依序為 a_1, a_2, \dots, a_N 。
- 決定一個神祕數字 k ，然後對於每個點 a_i 計算出該點與 $a_{i+1}, a_{i+2}, \dots, a_{i+k}$ 這些點中的最近點對距離 b_i ，接著我們便說最小的 b_i 就是我們找到的 N 個點中的最近點對。

正式來說，報紙上的做法說要計算：

$$\min_{i=1}^N \left(\min_{j=i+1}^{\min(j+k, N)} Dis(a_i, a_j) \right)$$

你想了一陣子之後覺得，小孩子才看報紙，上面的一定是錯的！因此請你構造一份測資讓底下的程式碼出錯。

報紙上還附了一份實作的程式碼，最後會輸出最近點對距離的平方。

附帶一提，底下的程式碼 **正確的** 實作了上面提到的做法，並不需要尋找實作上的錯誤。另外仔細觀察後便會發現，並不存在一個測資 **一定** 會讓報紙上的做法出錯，但可以讓報紙上的做法 **高機率** 的出錯，本題會有 10 筆測資，只要在 10 筆測資中都讓底下的程式碼出錯即可 **AC**，本題不會無故 Rejudge。程式碼中

`chrono::steady_clock::now().time_since_epoch().count()` 是一個精度為 10^{-9} 秒的時間函數，在程式中作為 random seed，確保每次 judge 都會旋轉不同的角度。

- Hard version:
 - $k = 200$

Original Sample Input

```
2
0 0
0 2
```

Original Sample Output

```
4
```

Program To Be Hacked

```
#include <bits/stdc++.h>
using namespace std;
#define maxn 200005

int n;
pair<int, int> p[maxn];
long double angle, cos_v, sin_v;

bool cmp(pair<int, int> a, pair<int, int> b) {
    auto v1 = make_pair(a.first * cos_v - a.second * sin_v
        , a.first * sin_v + a.second * cos_v);
    auto v2 = make_pair(b.first * cos_v - b.second * sin_v
        , b.first * sin_v + b.second * cos_v);
    return v1 < v2;
}

int main() {
    cin.tie(0), cout.sync_with_stdio(0);
    mt19937 rng(chrono::steady_clock::now().time_since_epoch().count()); // random number generator
    cin >> n;
    for (int i = 1; i <= n; ++ i)
        cin >> p[i].first >> p[i].second;

    long long ans = 6e18, k = 200;
    angle = rng();
    cos_v = cos(angle);
    sin_v = sin(angle);
    sort(p + 1, p + 1 + n, cmp);
    for (int i = 1; i <= n; ++ i) { // Sweep backward k points
        for (int j = i + 1; j <= i + k && j <= n; ++ j) {
            long long v1 = (p[i].first - p[j].first);
            long long v2 = (p[i].second - p[j].second);
            long long dis = v1 * v1 + v2 * v2;
            ans = min(ans, dis);
        }
    }
    cout << ans << "\n";
    return 0;
}
```

Input Format

本題沒有輸入。

Output Format

第一行請輸出一個數字 N ，代表點數。
接下來一共有 N 行，請輸出兩個數字 x_i, y_i 代表第 i 個點的座標。

- $2 \leq N \leq 2 \times 10^5$
- $|x_i|, |y_i| \leq 10^9$

Sample Input

copy

Sample Output

copy

Hints

Sample Code

以下是一個格式正確但不會 **AC** 的範例程式

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    puts("2");
    puts("0 0");
    puts("0 1");
    return 0;
}
```

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0~9	100

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	5000	262144	65536	1	
1	5000	262144	65536	1	
2	5000	262144	65536	1	
3	5000	262144	65536	1	
4	5000	262144	65536	1	
5	5000	262144	65536	1	
6	5000	262144	65536	1	
7	5000	262144	65536	1	
8	5000	262144	65536	1	
9	5000	262144	65536	1	

9. 璃奈的煩惱 (駭客題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

66.7% (2/3)

Submission's AC Ratio

38.5% (5/13)

Tags

Description

Original Description

高咲侑是虹咲學園普通科的一員，自從某次在路上看到優木雪菜的表演後，就對學園偶像充滿著無限的憧憬，於是他決定鼓勵青梅竹馬上原步夢成為學園偶像，並與她一起參加自己學校的學園偶像同好會。雖然侑醬對學園偶像有著憧憬，但是她的願望是能夠應援偶像，因此在學園偶像同好會中她擔任的角色是負責幫助大家完成夢想，不論是給予其他成員擔任偶像的自信、剪輯表演影片或是研究其他偶像等等她都會幫忙大家。

這一次侑醬在研究前輩μ's的表演「SUNNY DAY SONG」時，突然很好奇是怎樣的編舞才能讓當時接近全國的學園偶像都一起表演的呢，於是她想方設法從各種地方蒐集了所有有關「SUNNY DAY SONG」的資料，研究完舞蹈動作後她大概理解了原因，但是對於所有人的站位安排她還是很困惑，想知道究竟兩個人之間的最近距離要多遠才能避免跳舞的時候互相干擾到呢。

然而那次的表演實在太多人了，加上侑醬還有其他的事情要處理，於是她希望有人能幫她分析出那次表演中站位的最近距離，但是她怕會有競爭對手趁機來偷取她辛辛苦苦蒐集來的情報，於是你必須提供她一個種子讓她指派某個時間點的站位給你來分析。

具體來說，她寫了一個如下的函數來根據你給她的種子選擇要給你哪個時間點的站位

```
void generate_problem(std::seed_seq &&);
```

而你則除了需要呼叫該函數來獲得站位分析外，也需要實作一個如下的函數來分析站位

```
double closest_pair(std::vector<std::pair<double, double>>);
```

Original Input Format

本題沒有輸入，如果你輸入任何東西，可能會導致各種不可預期的結果(?)。

請 `#include "libXXXX.h"` 之後實作下列函數，如果你的函數名稱不對或者長得不像下面那行，你將會獲得一個CE。

```
double closest_pair(std::vector<std::pair<double, double>>);
```

保證僅會呼叫一次此函數且呼叫時傳入 `closest_pair` 的 `std::vector` 大小 (N) 不會太大，且每個點 (x_i, y_i) 的範圍也在合理的範圍內。

- $3 \leq N \leq 10^5$
- $-10^9 \leq x_i, y_i \leq 10^9$

Original Output Format

本題沒有輸出

Original Limits

Time Limit: 1 second
Memory Limit: 64 MB

Judge Method

身為虹咲學園學園偶像同好會技術擔當的天王寺璃奈想當然決定協助侑醬完成此任務，而且為了讓自己的程式不要跑得太慢，璃奈徹夜學習了期望時間複雜度 $O(N)$ 的最近距離對演算法，並且寫完後交給侑醬。然而怕青梅竹馬被搶走的步夢這時突然出現，並表示璃奈的這個程式有可能跑得非常慢，要侑醬不要接受它。傷心的璃奈於是找上了你，希望你能告訴她怎樣的情況會出現問題呢？身為資深璃奈推的你，於是決定幫助她，產生出一個會讓璃奈的程式執行超過一秒的測資。

Program To Be Hacked

這裡 是你需要 hack 的程式，要 hack 此程式，你必須引入此檔案，並撰寫如下的函數

```
void generate_problem(std::seed_seq &&);
```

(該份程式碼的全域變數名稱 `rnd` 在原程式中並非為此命名，你也不需要使用 `rnd` 這個物件即可完成此題)

Input Format

本題請用 C++14/17 作答。

本題沒有輸入，如果你輸入任何東西，可能會導致各種不可預期的結果(?)。

請 `#include "lib0009.h"` 之後實作下列函數，如果你的函數名稱不對或者長得不像下面那行，你將會獲得一個CE。

```
void generate_problem(std::seed_seq &&);
```

注意：如果你在程式裡實作了main()函式，你也會獲得一個CE。

Sample Input

copy

Hints

這份程式 是你向侑醬要來的程式，因此此程式能產出合乎規範的測資，但不保證能讓璃奈的程式執行超過一秒鐘。

若是你的程式產生的測資不符合條件或是無法讓璃奈的程式執行超過一秒鐘，你可能會拿到一個 WA 。

如果有任何程式碼看不懂，或許可以參閱 [cppreference](#)

Output Format

本題沒有輸出

Sample Output

copy

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0	100

Testdata and Limits					⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1500	65536	65536	<div>1</div>	

✓ 10. 有錢人小 Y 過馬路 (通靈題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



王禴立、陳泰穎、
楊翔宇

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

50.0% (4/8)

Tags

Description

有天，小 Y 在去吃拉麵的路上看到了一些分隔島。緊接著，他看到了每個分隔島上都有一個警告標示，寫著的內容都大同小異，例如「違法穿越需罰款新台幣 1,200 元」之類的內容。飢餓的小 Y 看著馬路對面的拉麵店，再看看馬路上的分隔島，開始思考，如果小 Y 想要盡快到達拉麵店，他要走多遠才能到達？

我們可以把小 Y 的位置跟拉麵店的位置分別看成平面上的點，而每個馬路上的分隔島是一個平面上的線段。已知如果小 Y 決定從 P 點走到 Q 點，那麼小 Y 走的距離就是 $\sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2}$ 。

Input Format

第一行是一個整數 N ，代表分隔島的數量。接下來的 N 行，每行都有四個整數 s_x, s_y, t_x, t_y ，代表分隔島的兩個端點分別是 (s_x, s_y) 及 (t_x, t_y) 。最後一行有四個整數 S_x, S_y, T_x, T_y ，代表小 Y 現在在點 (S_x, S_y) ，且拉麵店在點 (T_x, T_y) 。

- $0 \leq N \leq 1000$
- 所有的座標範圍介於 $[0, 10^7]$

Output Format

輸出一個浮點數，為小 Y 要走的最短距離。只要輸出的答案與正確答案的絕對誤差或相對誤差在 10^{-9} 以內就可以被視為正確。

Sample Input

copy

0
1 1 2 2

Sample Output

copy

1.414213562373

Hints

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0~24	1

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1000	262144	65536	1	
1	1000	262144	65536	1	
2	1000	262144	65536	1	
3	1000	262144	65536	1	
4	1000	262144	65536	1	
5	1000	262144	65536	1	
6	1000	262144	65536	1	
7	1000	262144	65536	1	
8	1000	262144	65536	1	
9	1000	262144	65536	1	
10	1000	262144	65536	1	
11	1000	262144	65536	1	
12	1000	262144	65536	1	
13	1000	262144	65536	1	
14	1000	262144	65536	1	
15	1000	262144	65536	1	
16	1000	262144	65536	1	
17	1000	262144	65536	1	
18	1000	262144	65536	1	
19	1000	262144	65536	1	
20	1000	262144	65536	1	
21	1000	262144	65536	1	
22	1000	262144	65536	1	
23	1000	262144	65536	1	
24	1000	262144	65536	1	

✔ 11. 吐鈔機 (通靈題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

50.0% (1/2)

Tags

Description

這天，小 Y 發現了一台有趣的機器。經過小 Y 的研究，他發現這是一台吐鈔機，然而要讓這台機器吐鈔並不是什麼很簡單的事情。這台機器在啟動的時候會隨機的選取一個 N ，然後吐出 N 個隨機的整數 a_1, a_2, \dots, a_N 。只要使用者能夠在接下來的 10 秒以內輸入最多能夠有幾組元素皆在 1 到 N 的集合 $S = \{s_1, s_2, \dots s_{|S|}\}$ 滿足 a_{s_1} and a_{s_2} and \dots and $a_{s_{|S|}}$ (and 為位元 and) 皆不相同，這台機器就會開始吐出足以讓人支付數不盡的數量的罰單的鈔票。

集合可能為空的，這個情況之下這個集合的所有元素的 and 就是 0。

小 Y 發現他蠻需要這台機器吐出一些鈔票的，於是他請你幫他寫出一個程式，可以算出這台機器所要求的答案。

此題為互動題。你至少需要完成一個函數 `int64_t solve(std::vector<int>)`，其中參數即為這台機器吐出的整數，而回傳值則是這台機器所預期的答案，且請在檔案的最前端引入 `lib0011.h` 這個檔案。你不需要完成 `main` 函數，否則你可能會獲得一個 `Compile Error`，也不應該在 `solve` 函數中操作不是你自己宣告的全域變數，否則可能會發生各種不可預期的結果。

Input Format

此題沒有輸入。

- $1 \leq N \leq 35$
- 對所有合法的 i ， $1 \leq a_i \leq 2^{20}$

Output Format

此題沒有輸出。

Sample Input

copy

Sample Output

copy

Hints

這是一份在此題可以順利執行，編譯，且有高機率會獲得 Wrong Answer 的程式碼：

```
#include "lib0011.h"
#include <bits/stdc++.h>
using namespace std;

int64_t solve(vector<int> ans) {
    return 87;
}
```

Problem Source

Subtasks		
No.	Testdata Range	Score
1	0~4	1

Testdata and Limits					⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	1900	65536	65536	①	
1	1900	65536	65536	①	
2	1900	65536	65536	①	
3	1900	65536	65536	①	
4	1900	65536	65536	①	

✓ 12. Deja Vu? (通靈題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



王榆立、陳泰穎、楊翔宇

User's AC Ratio

60.0% (3/5)

Submission's AC Ratio

26.8% (11/41)

Tags

Description

本題請用 C++14/17 作答。

西元 2020 年 12 月的某一天，臺灣的競程選手們因故得知了 **loLive 所屬的夏色まつり擅長破譯密碼的這件事。

然而，告訴大家這一點的那位祭絲表示，他實在編不下去子不願意透露更多祕密了，於是此處我們就省略故事吧。

給定 64-bit 無號整數的排列 (permutation) τ 、一個 64-bit 無號整數 y ，以及一個能夠查詢 $\tau(x)$ 的標頭檔，請你找出一個 x 使得 $\tau(x) = y$ 。

Input Format

一個 64-bit 無號整數 y 。

Output Format

一個 64-bit 無號整數 x ，使得 $\tau(x) = y$ 。

Sample Input

copy

10525914753111268782

Sample Output

copy

7810209329417266199

Hints

標頭檔使用方式：

```
#include "lib0012.h"
// ...
uint64_t y = Magic(x); // Returns tau(x)
```

Problem Source

炒冷飯

Subtasks		
No.	Testdata Range	Score
1	0~5	100

Testdata and Limits					⌵
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks	
0	100	65536	65536	1	
1	100	65536	65536	1	
2	100	65536	65536	1	
3	100	65536	65536	1	
4	100	65536	65536	1	
5	1000	65536	65536	1	

13.Nekomancer (通靈題)


Submit

Status

Ranklist

Back to Problems List

TopCoder



admin

User's AC Ratio

100.0% (3/3)

Submission's AC Ratio

12.0% (6/50)

Tags

Description

「咚！」
「又在看其他女人的直播了嗎？」
「沒辦法，るしあ只好把你監禁起來了呢。」

雖然るしあ平時會無微不至地照顧你，但她畢竟有要工作的時候，此時無法使用網路（以免你花心）的你就會變得有些無聊。最近，你爲了這種時候發明了一個遊戲。（你被監禁在的）るしあの地下室有一排很長的瓷磚，上面有一些深淺不一看起來有點像血跡的東西（不過你暫時不去想那是什麼）。每個回合，你會找到不同的 v, l, r, c, k ，然後試圖以最快的速度回答，在 $[l, r]$ 之間是否有連續 k 塊瓷磚使得剛好有 c 塊的深淺度小於等於 v 。

不過，今天你忽然意識到，或許可以寫個程式解決這個問題。於是就很 anti-climatic 地開始寫了。

Input Format

第一行是一個正整數 n ，表示這一排瓷磚的塊數。

接下來的一行有 n 個空白分隔的正整數 a_i ，表示該塊瓷磚的深淺度。

再來的一行是一個正整數 q ，表詢問的回合數。

然後有 q 行，每行依序是空白分隔的 v, l, r, c, k ，其意義如題敘所述。

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 2 \times 10^5$
- $1 \leq a_i \leq 10^9$
- $1 \leq v \leq 10^9$
- $1 \leq l \leq r \leq n$
- $0 \leq c \leq 10^6$
- $1 \leq k \leq 10^6$

Output Format

對於每筆詢問請輸出一行，如果有這樣的連續 k 塊瓷磚請輸出 **1**，否則請輸出 **0**。

Sample Input

```
10
4 17 19 20 22 24 26 11 16 13
5
15 6 10 1 3
1 7 7 0 1
29 3 3 1 2
27 1 8 1 3
10 6 6 1 2
```

Sample Output

```
1
1
0
0
0
```

Hints

- 據說因爲瓷磚很小，所以電腦可以一次看到很多塊。
- <https://gcc.gnu.org/onlinedocs/gcc/Vector-Extensions.html>
- <https://db.in.tum.de/~finis/x86-intrin-cheatsheet-v2.2.pdf?lang=en>
- <https://software.intel.com/sites/landingpage/IntrinsicsGuide/>
- 你大概只需要第一個連結？
- Judge 支援 AVX2 和 AVX512。
- `#pragma GCC target("avx,avx2,avx512bw,avx512cd,avx512f,avx512dq,avx512vl,avx512vnni")`

Problem Source

炒冷飯

Subtasks				
No.	Testdata Range		Score	
Testdata and Limits				⬆
No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	2500	65536	65536	
1	2500	65536	65536	
2	2500	65536	65536	
3	2500	65536	65536	
4	2500	65536	65536	
5	2500	65536	65536	
6	2500	65536	65536	
7	2500	65536	65536	
8	2500	65536	65536	
9	2500	65536	65536	
10	2500	65536	65536	
11	2500	65536	65536	
12	2500	65536	65536	
13	2500	65536	65536	
14	2500	65536	65536	
15	2500	65536	65536	
16	2500	65536	65536	
17	2500	65536	65536	
18	2500	65536	65536	
19	2500	65536	65536	

✓ 14. 順基因方程式 - Easy (通靈題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

1.9% (1/53)

Tags

Description

基因算法是一個以模擬基因演化過程求優化解的演算法，在正常狀況下愈後面的子代平均表現愈好。

現在有一個演化過程中的群體，他們朝著一個共同的目標函數演化(每一個子題的目標函數都不相同)，每一個子題會提供前 x 個子代的基因群體。

請輸出一個基因，他在這個目標函數下勝過第 y 個子代的平均。

對於Easy子題， $x = 18$ ， $y = 20$ ，請參考資料夾中的 gene_easy_generation_*

https://drive.google.com/drive/folders/1PCaGKd_4swl8VwHjVENP6FudPOpQY3G8?usp=sharing

在檔案中，每一行是一個基因，在Easy子題裡有200個基因，每個基因長度為100。

Input Format

輸入是一個 1，作為這個版本的代號。

Output Format

輸出一個字元皆為 *ATCG* 的字串，長度為100，且在目標函數下勝過第 y 子代的平均。

Sample Input

copy

Sample Output

copy

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
-----	----------------	-------

Testdata and Limits



No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	

Submit

Status

Ranklist

Back to Top

✓ 15. 順基因方程式 - Mid (通靈題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (1/1)

Submission's AC Ratio

100.0% (1/1)

Tags

Description

基因算法是一個以模擬基因演化過程求優化解的演算法，在正常狀況下愈後面的子代平均表現愈好。

現在有一個演化過程中的群體，他們朝著一個共同的目標函數演化(每一個子題的目標函數都不相同)，每一個子題會提供前 x 個子代的基因群體。

請輸出一個基因，他在這個目標函數下勝過第 y 個子代的平均。

對於Mid子題， $x = 8$ ， $y = 10$ ，請參考資料夾中的 gene_mid_generation_*

https://drive.google.com/drive/folders/1PCaGKd_4swl8VwHjVENP6FudPOpQY3G8?usp=sharing

在檔案中，每一行是一個基因，在Mid子題裡有200個基因，每個基因長度為100。

Input Format

輸入是一個 2，作為這個版本的代號。

Output Format

輸出一個字元皆為 *ATCG* 的字串，長度為100，且在目標函數下勝過第 y 子代的平均。

Sample Input

copy

Sample Output

copy

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
-----	----------------	-------

Testdata and Limits



No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	

Submit

Status

Ranklist

Back to Top

✓ 16. 順基因方程式 - Hard (通靈題)

Submit

Status

Ranklist

Back to Problems List

TopCoder



zck 至高無上

zck 至高無上

User's AC Ratio

100.0% (2/2)

Submission's AC Ratio

100.0% (3/3)

Tags

Description

基因算法是一個以模擬基因演化過程求優化解的演算法，在正常狀況下愈後面的子代平均表現愈好。

現在有一個演化過程中的群體，他們朝著一個共同的目標函數演化(每一個子題的目標函數都不相同)，每一個子題會提供前 x 個子代的基因群體。

請輸出一個基因，他在這個目標函數下勝過第 y 個子代的平均。

對於Hard子題， $x = 12$ ， $y = 12$ ，請參考資料夾中的 gene_hard_generation_*

https://drive.google.com/drive/folders/1PCaGKd_4swl8VwHjVENP6FudPOpQY3G8?usp=sharing

在檔案中，每一行是一個基因，在Hard子題裡有200個基因，每個基因長度為100。

Input Format

輸入是一個 3，作為這個版本的代號。

Output Format

輸出一個字元皆為 *ATCG* 的字串，長度為100，且在目標函數下勝過第 y 子代的平均。

Sample Input

copy

Sample Output

copy

Hints

Problem Source

Subtasks

No.	Testdata Range	Score
-----	----------------	-------

Testdata and Limits



No.	Time Limit (ms)	Memory Limit (KiB)	Output Limit (KiB)	Subtasks
0	1000	262144	65536	

Submit

Status

Ranklist

Back to Top