



دانشکده مهندسی کامپیوتر

ساختمان های داده

امتحان عملی اول

تهیه و تنظیم سوالات:
مبین داریوش همدانی
بابک بهکام کیا

استاد درس: سید صالح اعتمادی

نیم سال اول ۱۴۰۱-۱۴۰۰

fb_E1	نام شاخه
E1	نام پروژه/پوشه/پول ریکوست
۱۷ آبان ساعت ۱۴:۳۰	مهلت ارسال

راهنمای تمرین

ساخت پروژه

۱. با اجرای اسکریپت ساخت پروژه در ریشه ریپازیتوری خود برای درس ساختمان داده، یک پروژه برای امتحان عملی با نام E1 بسازید:^۱

```
dsproj -Create `
-cname E1 `
-testcommon .\TestCommon\TestCommon.csproj `
-testdata .\DS_E1\_publish\TestData\
```

پارامترهای قرمز به ترتیب محل پروژه TestCommon و محل فولدر TestData امتحان را مشخص می‌کنند. این دو پارامتر را با توجه به موقعیت آنها بر روی کامپیوتر خود تغییر دهید.

۲. با اجرای دستورات زیر فایل مسائل (و Processor) را به پروژه خود اضافه کنید:

```
dsproj -Add .\E1\E1.sln -Problem .\DS_E1\Q1Partition.cs
```

```
dsproj -Add .\E1\E1.sln -Problem .\DS_E1\Q2Cars.cs
```

```
dsproj -Add .\E1\E1.sln -Problem .\DS_E1\Q3TeamSeas.cs
```

```
dsproj -Add .\E1\E1.sln -Problem .\DS_E1\E1Processors.cs
```

پارامتر بعد از Add- فایل سولوشن تمرین را مشخص می‌کند. پارامترهایی که با قرمز مشخص شده اند نیز فایل کلاس هر سوال را مشخص می‌کند، این پارامترها را با توجه به موقعیت آن بر روی کامپیوتر خود تغییر دهید.

۳. مشابه دستور قبل، فایل تست را با دستور زیر به پروژه خود اضافه کنید:

```
dsproj -Add .\E1\E1.sln -Test .\DS_E1\GradedTests.cs
```

سایر نکات

۱. استفاده از اینترنت فقط و فقط برای استفاده از نرم‌افزار Teams و درست کردن PullRequest و جستجو در مورد سینتکس یا Error Code مجاز است. چنانچه اشکالی در مورد امتحان داشتید با استاد/حل‌تمرین در محیط تیمز در میان بگذارید.

۲. استفاده از اسلایدهای درس و کدهایی که «خود شما» برای «این درس» نوشته و در گیت موجود دارید مجاز است. استفاده از هرگونه کد دیگر که یا توسط شما نوشته نشده یا در برای این درس نوشته نشده یا در گیت شما قبلاً موجود نبوده مجاز نمی‌باشد.

۳. استفاده از هرگونه ویدیو مجاز نمی‌باشد.

۴. تصویر صفحه نمایش و وبکم شما در کل مدت امتحان بدون وقفه باید توسط نرم‌افزار FlashBackExpress (یا نرم‌افزار مشابه) ضبط شده و پس از فشرده‌سازی برای استاد درس ارسال شود.

^۱می‌توانید به صورت دستی نیز مراحل ساخت پروژه را انجام دهید. در این صورت حتماً به نام پروژه و پوشه‌ها توجه داشته باشید که درست باشند.

۱ Partition (۲۵ نمره)

در دانشکده کامپیوتر n دانشجو وجود دارد. هر کدام از این دانشجویان یک امتیاز درسی دارند. امتیاز دانشجوی i ام با p_i نمایش داده می‌شود. آموزش دانشکده قصد دارد دانشجویان را به تعدادی گروه تقسیم کند به نحوی که اختلاف امتیاز درسی دانشجویان موجود در یک گروه حداکثر x باشد. برنامه‌ای بنویسید که حداقل تعداد گروه ممکن را محاسبه کند.

محدودیت‌ها و ورودی

در خط اول ورودی به ترتیب دو عدد n و x آمده‌اند. در خط دوم n عدد آمده است که امتیاز دانشجویان را مشخص می‌کند.

$$1 \leq n, x \leq 10^5 \bullet$$

$$1 \leq p_i \leq 10^4 \bullet$$

۱ نمونه

ورودی:

```
8 4
1 2 3 21 7 12 14 21
```

خروجی:

```
4
```

توضیح: در این مثال می‌توان گروه‌هایی با امتیازات زیر داشت:

$$۱. \{1, 2\}$$

$$۲. \{7, 3\}$$

$$۳. \{12, 14\}$$

$$۴. \{21, 21\}$$

۲ نمونه

ورودی:

```
6 2
12 15 7 8 19 24
```

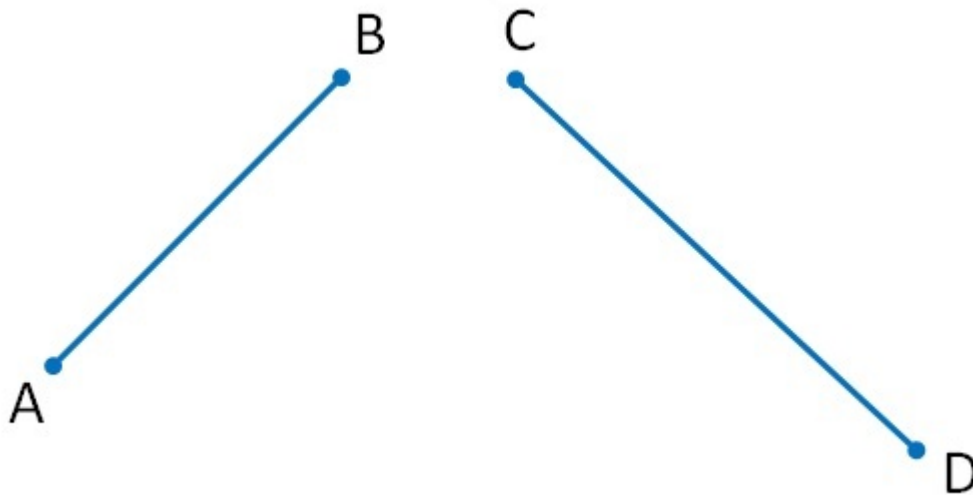
خروجی:

```
5
```

```
۱ using System;
۲ using TestCommon;
۳
۴ namespace E1
۵ {
۶     public class Q1Partition : Processor
۷     {
۸         public Q1Partition(string testDataName) : base(testDataName)
۹         {
۱۰         }
۱۱
۱۲         public override string Process(string inStr)
۱۳             => E1Processors.ProcessQ1Partition(inStr, Solve);
۱۴
۱۵         public long Solve(long n, long x, long[] p)
۱۶         {
۱۷             throw new NotImplementedException();
۱۸         }
۱۹     }
۲۰ }
```

۲ Cars (۳۵ نمره)

دو ماشین در صفحه مختصات دو بعدی قرار دارند. در لحظه صفر ماشین اول در نقطه $A = (a_x, a_y)$ قرار دارد و ماشین دوم در نقطه $C = (c_x, c_y)$ قرار دارد. در یک لحظه هر دو ماشین با هم شروع به حرکت می‌کنند. به طور دقیقتر، ماشین اول از نقطه A با سرعتی ثابت به سمت نقطه $B = (b_x, b_y)$ می‌رود و ماشین دوم از نقطه C با سرعتی ثابت به سمت نقطه $D = (d_x, d_y)$ می‌رود. علاوه بر این راننده‌های هر دو ماشین سرعت خود را به نحوی انتخاب می‌کنند که دقیقا در یک زمان و با هم به انتهای مسیرشان برسند. با گرفتن مختصات نقاط مبدا و مقصد هر دو راننده، کمترین فاصله بین دو ماشین را در طول زمان حساب کنید.



محدودیت‌ها و ورودی

در تنها خط ورودی به ترتیب اعداد طبیعی $a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y$ آمده اند که مختصات نقاط مبدا و مقصد هر دو راننده را مشخص می‌کنند. در خروجی کمترین فاصله بین دو ماشین در طول سفرشان را برگردانید. جواب خروجی شما درست محسوب می‌شود اگر اختلاف آن با جواب درست کمتر از 10^{-6} باشد.

$$0 \leq a_x, a_y, b_x, b_y, c_x, c_y, d_x, d_y \leq 10^5 \bullet$$

۱ نمونه

ورودی:

0 5 0 0 4 0 0 0

خروجی:

0.000000

توضیح: در این مثال هر دو ماشین در انتها به نقطه مبدا می‌رسند بنابراین کمترین فاصله برابر صفر است.

راهنمایی

۱. نمودار فاصله بین دو ماشین را بر حسب درصد زمان طی شده رسم کنید

۲. الگوریتم تقسیم و حلی از مرتبه $O(\log_3 n)$ با ایده‌ای مشابه Binary Search طراحی کنید که نقطه جواب را پیدا کند.

کلاس مسئله

```
۱ using System;
۲ using TestCommon;
۳
۴ namespace E1
۵ {
۶     public class Q2Cars : Processor
۷     {
۸         public Q2Cars(string testDataName) : base(testDataName)
۹         {
۱۰         }
۱۱
۱۲         public override string Process(string inStr)
۱۳             => E1Processors.ProcessQ2Cars(inStr, Solve);
۱۴
۱۵         public double Solve(
۱۶             long aX, long aY, long bX, long bY,
۱۷             long cX, long cY, long dX, long dY
۱۸         ) {
۱۹             throw new NotImplementedException();
۲۰         }
۲۱     }
۲۲ }
```

۳ TeamSeas # (۴۰ نمره)

یک شرکت که در زمینه پاکسازی اقیانوس‌ها فعالیت می‌کند، قصد دارد تا انتهای امسال مقدار زیادی زباله را از دریاها و اقیانوس‌ها خارج کند. این شرکت بودجه محدودی معادل m دلار را برای این عملیات اختصاص داده است.

جیمی (Jimmy) که مسئول مالی این شرکت است قصد دارد که با استفاده از بودجه موجود و برنامه‌ریزی مناسب، تا انتهای امسال بیشترین مقدار زباله را از اقیانوس خارج کند.

در مجموع n ناحیه از اقیانوس‌های سراسر جهان مشخص شده است که در آن زباله وجود دارد. به طور دقیق‌تر در ناحیه i ام به مقدار g_i کیلوگرم زباله وجود دارد. برای انجام پاکسازی در هر ناحیه دو روش وجود دارد:

۱. بستن قرارداد با پیمانکار مربوط به ناحیه i ام با هزینه c_i دلار. در این حالت پیمانکار قبول می‌کند که در ازای مبلغ پرداختی، به اندازه b_i کیلوگرم زباله را از آن ناحیه خارج کند.

۲. روش دیگر استخدام افراد محلی برای پاکسازی به صورت دستی است. در حوالی ناحیه i تعداد p_i نفر زندگی می‌کنند که هر کدام از آنها حاضرند در ازای دریافت s_i دلار، یک کیلوگرم زباله را جمع‌آوری کنند. علاوه بر این فرض می‌شود که هر فرد «توانایی» ثابتی برابر 10 کیلوگرم دارد و بیشتر از این مقدار را حتی در صورت وجود زباله مازاد نمی‌تواند جمع‌آوری کند. امکان خارج کردن مقدار اعشاری زباله نیز وجود ندارد (برای مثال $1/5$ کیلوگرم)

برنامه‌ای بنویسید که حداکثر مقدار زباله قابل جمع‌آوری را محاسبه کند.

محدودیت‌ها و ورودی

در خط اول دو عدد n و m آمده است که به ترتیب تعداد ناحیه‌ها و بودجه موجود را مشخص می‌کنند. در هر یک از پنج خط بعدی n عدد آمده است. این خطوط به ترتیب مقادیر زباله موجود در هر ناحیه $(g_1 \dots g_n)$ ، هزینه قرارداد با پیمانکار هر ناحیه $(c_1 \dots c_n)$ ، مقدار زباله خارج شده توسط پیمانکار هر ناحیه $(b_1 \dots b_n)$ ، جمعیت موجود در هر ناحیه $(p_1 \dots p_n)$ و دستمزد مربوط به هر ناحیه $(s_1 \dots s_n)$ را مشخص می‌کنند.

$$1 \leq n \leq 100 \bullet$$

$$1 \leq m \leq 5000 \bullet$$

$$1 \leq g_i, c_i, b_i, s_i \leq 10^4 \bullet$$

$$1 \leq p_i \leq 10 \bullet$$

$$b_i \leq g_i \quad \forall 1 \leq i \leq n \bullet$$

نمونه ۱

ورودی:

```
5 49
50 50 50 50 50
10 13 12 8 8
20 30 20 18 10
1 1 3 1 1
4 4 1 10 3
```

خروجی:

94

کلاس مسئله

```
1 using System;
2 using TestCommon;
3
4 namespace E1
5 {
6     public class Q3TeamSeas : Processor
7     {
8         public Q3TeamSeas(string testDataName) : base(testDataName)
9         {
10         }
11
12         public override string Process(string inStr)
13             => E1Processors.ProcessQ3TeamSeas(inStr, Solve);
14
15         public long Solve(long n, long m, long[] g, long[] c,
16             long[] b, long[] p, long[] s)
17         {
18             throw new NotImplementedException();
19         }
20     }
21 }
```