

Investice do pozemků

Termín odevzdání:	04.12.2016 23:59:59
Hodnocení:	5.5000
Max. hodnocení:	5.0000 (bez bonusů)
Odevzdaná řešení:	2 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)
Nápovědy:	1 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)

Úkolem je vytvořit program, který bude vyhledávat optimální investice do pozemků.

Na vstupu programu je zadání cen parcel. Předpokládáme, že parcely leží v pravoúhlém rastru, kde známe počet řádek a sloupců. Cena je známá pro každou parcelu, jedná se o celé číslo (kladné, nulové, záporné). Záporná (nulová) cena parcely je přípustná, může se např. jednat o parcelu s břemenem. Po zadání cen pozemků následuje seznam dotazů. Chceme investovat zadaný objem peněz a hledáme parcelu/parcely, které mají v součtu cenu přesně rovnou zadanému objemu peněz. Jsme ale omezeni tím, že můžeme nakupovat pouze sousední parcely. Zakoupené parcely navíc musí tvořit obdélník či čtverec v rastru. Program dokáže zpracovávat dotazy dvou typů: buď pouze zobrazí počet různých možností, jak investovat zadanou částku (dotaz count), nebo navíc vypíše i seznam parcel, které dotazu vyhovují (dotaz list).

Vstupem programu je:

- velikost rastru (šířka, výška), velikost je omezena na 1 až 2000 v každém směru,
- ceny jednotlivých parcel, ceny jsou zadané po řádcích,
- seznam dotazů.

Dotaz je buď typu count x nebo list x , kde x je investovaná částka.

Výstupem programu je vyřešení dotazů:

- na dotaz typu count x je odpovědí počet různých způsobů, kterými lze investovat částku x ,
- na dotaz typu list x je odpovědí seznam alokací parcel, následovaný počtem nalezených alokací (tedy číslo stejné, jaké by vrátil dotaz typu count x). Seznam alokací parcel má podobu:

$$x @ (x_1, y_1) - (x_2, y_2)$$

kde x_1, y_1 je souřadnice levého horního rohu alokace parcel a x_2, y_2 je souřadnice pravého dolního rohu alokace parcel.

Pokud je vstup neplatný, program to musí detekovat a zobrazit chybové hlášení. Chybové hlášení zobrazujte na standardní výstup (ne na chybový výstup). Za chybu považujte:

- rozměr rastru je nečíslný, nulový, záporný nebo překračuje limit 2000,
- zadaná cena parcel není číslo,
- dotaz není typu count ani list,
- číslo v dotazu chybí / není správně zadané.

Před implementací programu si rozmyslete, jakým způsobem budete reprezentovat ceny pozemků a jak budete v zadaném rastru vyhledávat. Velikost rastru je omezená na max. 2000 prvků v každém směru. Pro řešení tedy postačuje staticky alokovaná paměť.

Vyhledávání v cenách pozemků může trvat velmi dlouho. Naivní řešení má složitost n^6 , vylepšováním algoritmu se dá složitost výrazně snížit. Časové limity testovacího prostředí jsou nastavené tak, aby rozumná implementace naivního algoritmu prošla všemi testy mimo testů bonusových.

Ukázka práce programu:

Velikost mapy:

5 6

Cenova mapa:

```
-13  9 -16  14   3
 11  7  -5  14   9
 -2  -5   9  -9  -4
 11 13   3   8 -16
```

```
-1 -15 1 -4 2
3 7 -8 0 27
```

Dotazy:

list 12

12 @ (0,3) - (2,4)

12 @ (0,4) - (4,5)

12 @ (2,2) - (2,3)

12 @ (4,0) - (4,1)

Celkem: 4

count 12

Celkem: 4

count 11

Celkem: 10

list 11

11 @ (0,1) - (0,1)

11 @ (0,3) - (0,3)

11 @ (0,2) - (0,5)

11 @ (0,1) - (1,2)

11 @ (0,2) - (1,5)

11 @ (0,2) - (3,5)

11 @ (1,0) - (1,2)

11 @ (1,1) - (3,2)

11 @ (2,2) - (3,3)

11 @ (2,3) - (3,3)

Celkem: 10

count -3

Celkem: 2

list -3

-3 @ (0,0) - (4,0)

-3 @ (2,4) - (3,4)

Celkem: 2

list 9

9 @ (0,1) - (0,2)

9 @ (0,2) - (0,3)

9 @ (0,0) - (0,5)

9 @ (0,0) - (2,5)

9 @ (0,2) - (3,4)

9 @ (1,0) - (1,0)

9 @ (1,0) - (1,4)

9 @ (1,2) - (4,5)

9 @ (2,2) - (2,2)

9 @ (2,1) - (3,1)

9 @ (2,1) - (3,2)

9 @ (2,1) - (3,5)

9 @ (2,0) - (4,4)

9 @ (2,1) - (4,3)

9 @ (2,2) - (4,5)

9 @ (3,1) - (3,4)

9 @ (3,1) - (3,5)

9 @ (4,1) - (4,1)

9 @ (4,2) - (4,5)

Celkem: 19

list 0

0 @ (1,1) - (1,4)

0 @ (1,2) - (1,5)

0 @ (1,0) - (2,5)

0 @ (1,2) - (3,5)

0 @ (2,1) - (2,5)

0 @ (2,2) - (3,2)

0 @ (2,2) - (3,5)

0 @ (2,3) - (3,5)

0 @ (3,5) - (3,5)

0 @ (3,1) - (4,4)

Celkem: 10

count 10000

Celkem: 0

Velikost mapy:

4 3

Cenova mapa:

1 -1 1 -1
-1 1 -1 1
1 -1 1 -1

Dotazy:

count 0

Celkem: 36

list 0

0 @ (0,0) - (1,0)
0 @ (0,0) - (3,0)
0 @ (2,0) - (3,0)
0 @ (1,0) - (2,0)
0 @ (0,0) - (0,1)
0 @ (0,0) - (1,1)
0 @ (1,0) - (1,1)
0 @ (0,0) - (2,1)
0 @ (1,0) - (2,1)
0 @ (2,0) - (2,1)
0 @ (0,0) - (3,1)
0 @ (1,0) - (3,1)
0 @ (2,0) - (3,1)
0 @ (3,0) - (3,1)
0 @ (0,0) - (1,2)
0 @ (0,0) - (3,2)
0 @ (2,0) - (3,2)
0 @ (1,0) - (2,2)
0 @ (1,1) - (2,1)
0 @ (0,1) - (1,1)
0 @ (0,1) - (3,1)
0 @ (2,1) - (3,1)
0 @ (0,1) - (0,2)
0 @ (0,1) - (1,2)
0 @ (1,1) - (1,2)
0 @ (0,1) - (2,2)
0 @ (1,1) - (2,2)
0 @ (2,1) - (2,2)
0 @ (0,1) - (3,2)
0 @ (1,1) - (3,2)
0 @ (2,1) - (3,2)
0 @ (3,1) - (3,2)
0 @ (0,2) - (1,2)
0 @ (0,2) - (3,2)
0 @ (2,2) - (3,2)
0 @ (1,2) - (2,2)

Celkem: 36

Velikost mapy:

1 12

Cenova mapa:

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

Dotazy:

count 0

Celkem: 0

list 9

9 @ (0,1) - (0,3)
9 @ (0,5) - (0,7)
9 @ (0,9) - (0,11)

Celkem: 3

count 10

Celkem: 9

list 10

10 @ (0,0) - (0,3)
10 @ (0,1) - (0,4)
10 @ (0,2) - (0,5)
10 @ (0,3) - (0,6)
10 @ (0,4) - (0,7)
10 @ (0,5) - (0,8)

```
10 @ (0,6) - (0,9)
10 @ (0,7) - (0,10)
10 @ (0,8) - (0,11)
Celkem: 9
buy 10
Nespravny vstup.
```

```
Velikost mapy:
2 2
Cenova mapa:
1 2 3 test
Nespravny vstup.
```

Poznámky:

- Znak odřádkování (`\n`) je i za poslední řádkou výstupu (i za případným chybovým hlášením).
- Pro reprezentaci cen parcel postačuje datový typ `int`.
- Souřadnice (0,0) odpovídá levému hornímu rohu mapy, souřadnice rostou směrem vpravo a směrem dolů.
- Dynamická alokace není v této úloze potřeba. Maximální velikost rastru je omezená, paměťové limity postačují pro statickou alokaci. Je ale možné, že se některé reprezentace cenové mapy nevejdou do lokálních proměnných. V takovém případě může být řešením cenovou mapu alokovat v datovém segmentu (hint: klíčové slovo `static`).
- Odřádkování v cenové mapě na vstupu může, ale nemusí respektovat velikost rastru. Program se při načítání vstupu nemusí odřádkováním zabývat, pro zpracování mu stačí dříve zadaná velikost rastru.
- Pro velké velikosti rastru v bonusových testech je nalezeno velké množství vyhovujících alokací. Samotný výpis těchto alokací je dost pomalý. Proto se v bonusových testech kontrolují pouze dotazy typu `count`.
- Pořadí vyhovujících alokací v dotazech typu `list` není určené. Vaše implementace může nalezené alokace vypisovat v libovolném pořadí, testovací prostředí si před porovnáním pořadí ve výpisu upraví. Tedy například pro zadání:

```
Velikost mapy:
4 3
Cenova mapa:
1 -1 1 -1
-1 1 -1 1
1 -1 1 -1
Dotazy:
list 0
```

jsou přípustné odpovědi:

```
0 @ (0,0) - (1,0)
0 @ (0,0) - (3,0)
0 @ (2,0) - (3,0)
0 @ (1,0) - (2,0)
0 @ (0,0) - (0,1)
0 @ (0,0) - (1,1)
0 @ (1,0) - (1,1)
0 @ (0,0) - (2,1)
0 @ (1,0) - (2,1)
0 @ (2,0) - (2,1)
0 @ (0,0) - (3,1)
0 @ (1,0) - (3,1)
0 @ (2,0) - (3,1)
0 @ (3,0) - (3,1)
0 @ (0,0) - (1,2)
0 @ (0,0) - (3,2)
0 @ (2,0) - (3,2)
0 @ (1,0) - (2,2)
0 @ (1,1) - (2,1)
0 @ (0,1) - (1,1)
0 @ (0,1) - (3,1)
0 @ (2,1) - (3,1)
0 @ (0,1) - (0,2)
0 @ (0,1) - (1,2)
0 @ (1,1) - (1,2)
0 @ (0,1) - (2,2)
0 @ (1,1) - (2,2)
0 @ (2,1) - (2,2)
```

```

0 @ (0,1) - (3,2)
0 @ (1,1) - (3,2)
0 @ (2,1) - (3,2)
0 @ (3,1) - (3,2)
0 @ (0,2) - (1,2)
0 @ (0,2) - (3,2)
0 @ (2,2) - (3,2)
0 @ (1,2) - (2,2)
Celkem: 36

```

nebo:

```

0 @ (0,0) - (0,1)
0 @ (0,0) - (1,0)
0 @ (0,0) - (1,1)
0 @ (0,0) - (1,2)
0 @ (0,0) - (2,1)
0 @ (0,0) - (3,0)
0 @ (0,0) - (3,1)
0 @ (0,0) - (3,2)
0 @ (0,1) - (0,2)
0 @ (0,1) - (1,1)
0 @ (0,1) - (1,2)
0 @ (0,1) - (2,2)
0 @ (0,1) - (3,1)
0 @ (0,1) - (3,2)
0 @ (0,2) - (1,2)
0 @ (0,2) - (3,2)
0 @ (1,0) - (1,1)
0 @ (1,0) - (2,0)
0 @ (1,0) - (2,1)
0 @ (1,0) - (2,2)
0 @ (1,0) - (3,1)
0 @ (1,1) - (1,2)
0 @ (1,1) - (2,1)
0 @ (1,1) - (2,2)
0 @ (1,1) - (3,2)
0 @ (1,2) - (2,2)
0 @ (2,0) - (2,1)
0 @ (2,0) - (3,0)
0 @ (2,0) - (3,1)
0 @ (2,0) - (3,2)
0 @ (2,1) - (2,2)
0 @ (2,1) - (3,1)
0 @ (2,1) - (3,2)
0 @ (2,2) - (3,2)
0 @ (3,0) - (3,1)
0 @ (3,1) - (3,2)
Celkem: 36

```

nebo libovolná další ze zbývajících 371993326789901217467999448150835199999998 permutací.

- Slovní popis struktury platných vstupních dat není zcela exaktní. Proto na výtky některých studentů přikládáme i popis vstupního jazyka v EBNF:

```

input      ::= { whiteSpace } gridSize { whiteSpace } priceMap { whiteSpace } queryList
whiteSpace ::= ' ' | '\t' | '\n' | '\r'
gridSize  ::= integer { whiteSpace } integer
priceMap  ::= integer { { whiteSpace } integer }
queryList ::= { query { whiteSpace } }
query     ::= ( 'list' | 'count' ) { whiteSpace } integer
integer   ::= ['+' | '-'] digit { digit }
digit     ::= '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'

```

Vzorová data:

Download

☐ Referenční řešení

2

27.11.2016 00:08:57

Download

Stav odevzdání: Ohodnoceno

Hodnocení: 5.5000

• Hodnotitel: automat

- Program zkompileován
- Test 'Základní test s parametry podle ukázky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Max doba běhu: 0.005 s (limit: 1.000 s)
 - Celková doba běhu: 0.018 s
 - Úspěch v závazném testu, hodnocení: 100.00 %
- Test 'Test mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.005 s (limit: 1.000 s)
 - Celková doba běhu: 0.029 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Kontrola ošetření vstupních dat': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.005 s (limit: 1.000 s)
 - Celková doba běhu: 0.074 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test náhodnými daty': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.377 s (limit: 2.000 s)
 - Celková doba běhu: 3.577 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test náhodnými daty + mem debugger': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.037 s (limit: 2.000 s)
 - Celková doba běhu: 0.342 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Bonus #1 - rychlost': Program překročil přidělenou maximální dobu běhu
 - Program násilně ukončen po: 2.003 s (limit: 2.000 s)
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
 - Překročení doby běhu [**Zpřístupnit nápovědu (2.00 KiB, pouze část dat)**]
- Test 'Bonus #2 - rychlost': Nebylo testováno
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
- Test 'Bonus #3 - rychlost': Nebylo testováno
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Použité nápovědy: 1
- Penalizace za vyčerpané nápovědy: Není (1 <= 2 limit)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.50
- Celkem bodů: 1.00 * (5.00 + 0.50) = 5.50

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	4	--	--	--
	Řádek kódu:	75	18.75 ± 4.97	26	processEstate(...)
	Cyklotrická složitost:	33	8.25 ± 2.38	11	main(void)

1

26.11.2016 22:46:02

Download

Stav odevzdání: Ohodnoceno

Hodnocení: 5.5000

• Hodnotitel: automat

- Program zkompileován
- Test 'Základní test s parametry podle ukázky': Úspěch
 - Dosaženo: 100.00 %, požadováno: 100.00 %
 - Max doba běhu: 0.004 s (limit: 1.000 s)
 - Celková doba běhu: 0.017 s
 - Úspěch v závazném testu, hodnocení: 100.00 %

- Test 'Test mezních hodnot': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.005 s (limit: 1.000 s)
 - Celková doba běhu: 0.027 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Kontrola ošetření vstupních dat': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.004 s (limit: 1.000 s)
 - Celková doba běhu: 0.071 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test náhodnými daty': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.365 s (limit: 2.000 s)
 - Celková doba běhu: 3.492 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Test náhodnými daty + mem debugger': Úspěch
 - Dosaženo: 100.00 %, požadováno: 20.00 %
 - Max doba běhu: 0.035 s (limit: 2.000 s)
 - Celková doba běhu: 0.318 s
 - Úspěch v nepovinném testu, hodnocení: 100.00 %
- Test 'Bonus #1 - rychlost': Program překročil přidělenou maximální dobu běhu
 - Program násilně ukončen po: 2.002 s (limit: 2.000 s)
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
 - ☐ Překročení doby běhu
- Test 'Bonus #2 - rychlost': Nebylo testováno
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
- Test 'Bonus #3 - rychlost': Nebylo testováno
 - Neúspěch v bonusovém testu, hodnocení: Bonus nebude udělen
- Všechny paměťové bloky byly uvolněné - ok.
- Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Bonus za včasné odevzdání: 0.50
- Celkem bodů: $1.00 * (5.00 + 0.50) = 5.50$

		Celkem	Průměr	Maximum	Jméno funkce
SW metriky:	Funkce:	4	--	-- --	
	Řádek kódu:	75	18.75 ± 4.97	26	processEstate(...)
	Cyklostatická složitost:	33	8.25 ± 2.38	11	main(void)