

108資料結構

第一次作業

小偷背包問題

- 從前有個小偷叫Bane.他蓄謀已久想盜竊一家珠寶店的珠寶,但只帶了一個背包,珠寶還都很大顆.所以他想在警察來之前帶走越多東西或者越貴的東西越好.

小偷背包問題-1 (20%)

- Bane前往珠寶店,但發現門被密碼鎖鎖住了,密碼是數字.他突然想起了他的好朋友Riddler曾經跟他講過的一段話”密碼是四維空間中的一個位置...”
- Input
 - 四個數字(都是int),中間以空格間隔,且皆小於7
- Output
 - 輸出密碼(int)
- 注意事項
 - Input中的四個數代表著一個大小為 $7*7*7*7$ 的四維array中的一個位置.這個位置的值固定為1.且此array是column major
 - 意思就像`int array[7][7][7][7]; array[2][4][1][6] = 1;`//2 4 1 6為input
 - 計算此位置在一維空間中的位置,再加上1.就像是`array[?]`,?就是結果,output是? $+1$
 - 四維array從`[0][0][0][0]`到`[6][6][6][6]`
 - 程式跑一次就可以
- 評分標準
 - 跑出來的結果對則分數全拿,若不對則完全沒分
 - 若有一部分對,一部分不對.則看程式碼,酌情給一半分或都不給

小偷背包問題-1 (20%)

example

- Simple Input

2 4 1 6

- Simple output

2138

小偷背包問題-2 (20%)

- 但是Bane發現他只對了前部分,後部分他不知道.他想起了Riddler那段話的後半段”...四維空間的大小不一定固定”
- Input
 - 第一列四個數字(都是int),中間以空格間隔,代表位置
 - 第二列與第一列數字一一對應(都是int),代表空間大小.且每一組對應數字皆是下大於上
- Output
 - 輸出密碼(int)
- 注意事項
 - 第一列Input中的四個數代表四維array中的一個位置.這個位置的值固定為1.且此array是row major
 - 第二列Input中的四個數代表四維array的大小.
 - 意思就像int array[5][9][3][7]; array[2][4][1][6] = 1;//2 4 1 6為第一列input, 5 9 3 7為第二列input
 - 計算此位置在一維空間中的位置,再加上1.就像是array[?],?就是結果,output是?+1
 - 四維array從[0][0][0][0]開始
 - 程式跑一次就可以
- 評分標準
 - 跑出來的結果對則分數全拿,若不對則完全沒分
 - 若有一部分對,一部分不對.則看程式碼,酌情給一半分或都不給

小偷背包問題-2 (20%)

example

- Simple Input

2 4 1 6

5 9 3 7

- Simple output

476

小偷背包問題-3 (20%)

- Bane準備要來偷東西了
- Input
 - 第一列含一個數是背包大小(背包大小必小於所有物品大小之總和) (int)
 - 第二列含一列數(含10個數,每個數以空格分開)是物品大小(int)
- Output
 - 輸出最多能放多少物品(所有珠寶的大小總和)(int)
 - 換行後輸出你選擇的物品的尺寸(int)以空格隔開
- 注意事項
 - 程式跑一次就可以
 - 要用到stack(或queue)
 - 物品限取一次!
- 評分標準
 - 跑出來的結果對則分數全拿,若不對則完全沒分
 - 若有一部分對,一部分不對.則看程式碼,酌情給一半分或都不給
 - 在使用到stack (或queue)那列以註解"//stack (//queue)"標註出來

小偷背包問題-3 (20%)

example

- Simple Input

100

78 25 36 4 11 44 500 38 101 98

- Simple output

99

38 36 25

//由38+36+25得到99

小偷背包問題-4 (20%)

- Bane覺得有些東西裝不下好可惜於是他想打碎了再拿走
- Input
 - 第一列含一個數是背包大小(背包大小必小於所有物品大小之總和) (int)
 - 第二列含一列數(含10個數,每個數以空格分開)是物品大小 (int)
 - 第三列含一列數(含10個數,每個數以空格分開)是物品價值與上一列逐一對應 (int)
- Output
- 輸出
 - 輸出可以拿出的最高的價格(int)
 - 換行後輸出你選擇的物品的大小(int)以空格隔開
- 注意事項
 - 物品可以拆碎帶走(比如說一個物品大小為5價值為10元,他可以帶走該物品一個單位的大小,得到2元)
 - 程式跑一次就可以
 - 計算過程中自動捨棄小數部分
 - 要用到stack (或queue)
 - 物品限取一次!
- 評分標準
 - 跑出來的結果對則分數全拿,若不對則完全沒分
 - 若有一部分對,一部分不對.則看程式碼,酌情給一半分或都不給
 - 在使用到stack (或queue)那列以註解"//stack (//queue)"標註出來

小偷背包問題-4 (20%)

example

- Simple Input

100

78 25 36 4 11 44 500 38 101 98

100 20 36 125 13 46 99 48 210 48

- Simple output

316

4 101

//由 $(125/4)*4+210/101*96 = 316$

小偷背包問題-5 (20%)

- 警察沒有趕來,Batman卻提前出現了.但擋在Batman面前是一條條詭異的道路,Batman計算走哪一條路可以最快抓到Bane
- Input
 - 是一個matrix(7*7)!數字與數字之間以空格隔開(int)
- Output
 - 最短時間(int),或者是0(代表bane無法被抓住)
 - Print出你的程式所用的時間複雜度上限以bigO表示(string)
- 注意事項
 - matrix中的 [0,0]點為Batman,[6,6]為Bane
 - 數字介於0~9之間0(代表此路不通),1~9(代表該路的時間權重)
 - 比如說matrix中[3,5]這個點是4.代表從點3到點5此路要走4sec
 - 若matrix中[5,3]為0.則代表Batman無法從5點走到3點
 - 程式跑一次就可以
 - 要在 $O(n^2)$ 的時間複雜度中算出
- 評分標準
 - 跑出來的結果對則分數全拿
 - 若最短時間對但時間複雜度超過得一半分
 - 若都不對則沒分

小偷背包問題-5 (20%)

example

- Simple Input

0 8 9 1 0 3 0

0 0 2 0 9 0 4

0 0 0 2 0 2 7

0 0 0 0 2 7 8

0 0 0 0 0 4 1

0 0 0 0 0 0 9

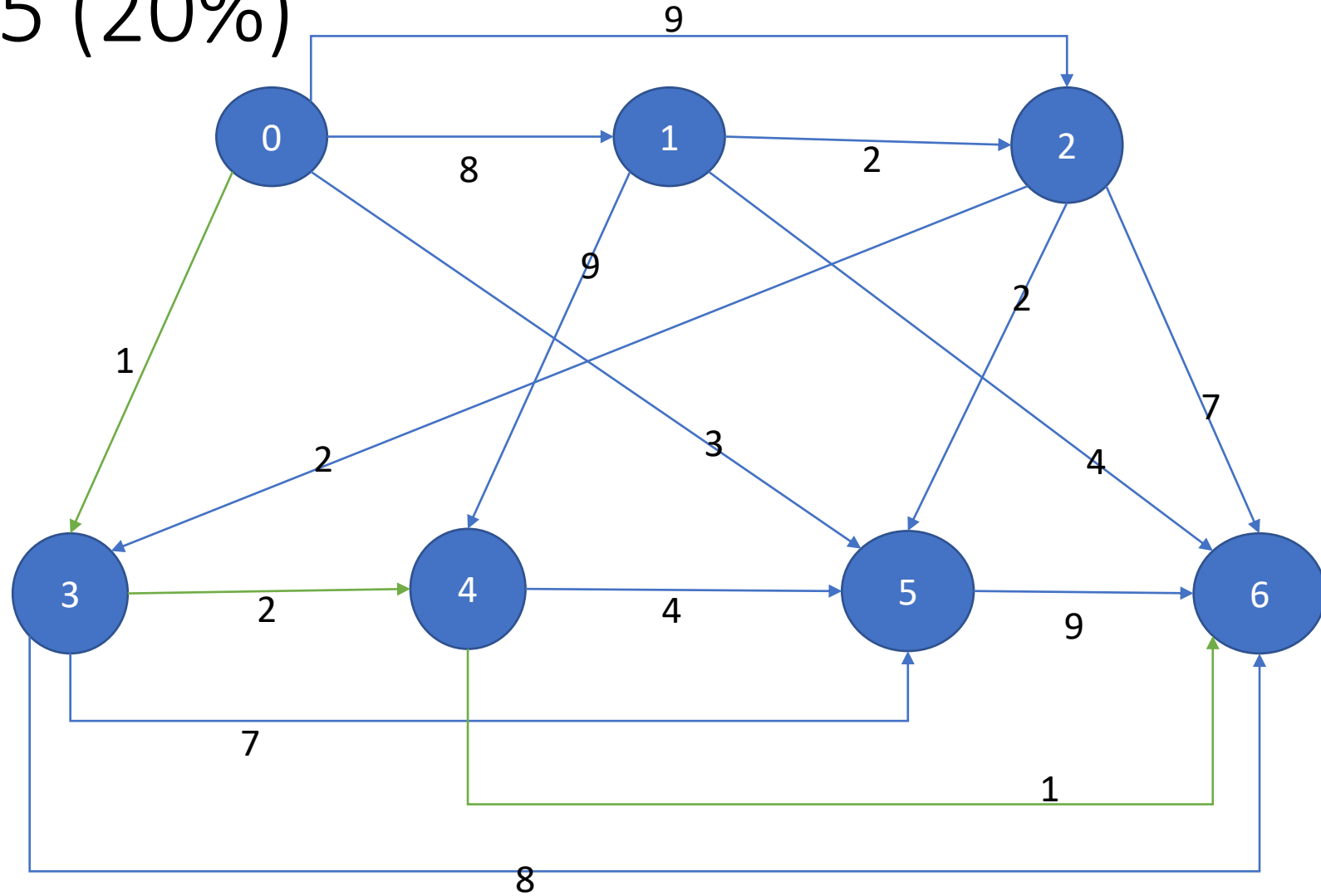
0 0 0 0 0 0 0

- Simple output

4

$O(n \log n)$

//由 $[0,3] + [3,4] + [4,6] = 1 + 2 + 1 = 4$



繳交方式

- Accepted C/C++
- Deadline : 2018/10/28(一) 中午12:00
- Input必須是read file!(input1.txt , input2.txt , ...)
- Output也必須是read file(output1.txt , output2.txt , ...)output檔不用繳交
- 檔名 : main.c若有很多檔案請標清楚題號比如,main1.c,main2.c,...
- 繳交方式:將你寫的所有檔案(包含main檔和input檔)放在一起壓縮,以i-learning方式繳交,檔名為“學號+姓名+homework1”.
- 禁止抄襲 , 可以的話打上註解方便閱讀
- 至少要跑出我給的測資出來,批改的時候還會測其它測資!
- 我們會以Dev-C++來做批改