THUẬT TOÁN THAM LAM

Bùi Đức Thiện – Chuyên Tin 2012-2016

Tầm nhìn ta thật ngắn mà đã thấy bao thứ để làm — Alan Turing

I/Giới Thiệu

"Đến đứa trẻ 3 tuổi biết tham"

Cũng giống quy hoạch động, tham lam là thuật tóan tối ưu hóa. Thuật tóan tham lam tiến hành trong nhiều bước. Tại mỗi bước nó đưa ra lựa chọn tốt nhất với hy vọng sẽ đóng góp một phần cho lời giải tối ưu toàn cục.

Thuật tóan tham lam có ưu điểm: mạnh mẽ, hiệu quả, dễ dàng trong cài đặt, chạy nhanh.

Tuy không phải lúc nào giải thuật tham lam cũng là giải thuật tối ưu, nhưng một khi đã chứng minh được thuật tóan tham lam là tối ưu, thì đây luôn là giải pháp được lựa chọn vì tốc độ và sự đơn giản trong việc code.

II/ Một số bài tóan dùng tham lam

1/ Bài tóan cái túi:

Bài tóan được phát biểu như sau:

Một kẻ trộm đột nhập vào một cửa hiệu tìm thấy có *n* mặt hàng có trọng lượng và giá trị khác nhau, nhưng hắn chỉ mang theo một cái túi có sức chứa về trọng lượng tối đa là *M*. Vậy kẻ trộm nên bỏ vào ba lô những món nào và số lượng bao nhiêu để đạt giá trị cao nhất trong khả năng mà hắn có thể mang đi được.

Ta có *n* loại đồ vật, x_1 tới x_n. Mỗi đồ vật x_j có một giá trị p_j và một khối lượng w_j. Khối lượng tối đa mà ta có thể mang trong ba lô là *M*.

Trong bài tóan này, ta cần tìm ra M kg giá trị nhất để mang đi, vậy hãy đơn giản, tính giá trị mỗi kg của 1 đồ vật, sau đó cố gắng mang đi M kg có giá trị lớn nhất. Ta sắp xếp lại toàn bộ theo tỉ lệ giá trị mỗi kg của các đồ vật. Sau đó , ta chỉ việc nhét đồ vật có giá trị mỗi kg lớn nhất nhất có thể đến khi cái túi đầy.

2/Bài toán đổi tiền

Bài tóan được phát biểu như sau:

Ở nước Vờ Nờ , Ai cũng muốn làm giàu, nên luôn cố gắng làm việc để được lương cao, tuy nhiên người ta cũng không hề muốn phải cầm nhiều tờ tiền vì rất dễ rơi. Mỗi lần đi lĩnh lương , mọi người luôn yêu cầu cầm số tờ tiền ít nhất. Đất nước có n loại tiền D1, D2, ... , Dn. Tính

xem, với 1 người có mức lương là M , khi nhận lương cần nhận tối thiểu bao nhiêu tờ tiền.

Lời giải sử dụng tham lam: Chọn đồng có mệnh giá lớn nhất không lớn hơn sô tiền cần đổi, sau đó lặp lại đến khi đạt được kết quả đề bài.

Thuật toán này không phải lúc nào cũng hoạt động hiệu quả:

VD: đổi tờ 8 đồng, trong cửa hàng có những mệnh giá 1,4,5. Theo thuật tóan tham lam, ta sẽ chọn 1 tờ 5 và 3 tờ 1. Nhưng thực tế chỉ cần chọn 2 tờ 4 là được kết quả.

Tuy nhiên, thuật tóan tham lam đạt kết quả tốt nhất ở mức thường xuyên, trên thực tế, với hệ thống tiền tệ thật sự hiện nay, thuật tóan này gần như không thể sai.

3/Bài toán lựa chọn hoạt động

Đi du lịch, L rất ham chơi, muốn chơi càng nhiều trò chơi càng tốt, tuy nhiên, các trò chơi tổ chức ở những khung thời gian khác nhau và thường xen lẫn vào thời gian của nhau(trò này chưa kết thúc thì trò khác đã bắt đầu), L không thể rời 1 trò chơi khi nó chưa kết thúc. Hãy giúp L chọn ra những trò chơi để tham gia sao cho chơi được nhiều nhất.

Ví dụ có 9 trò chơi:

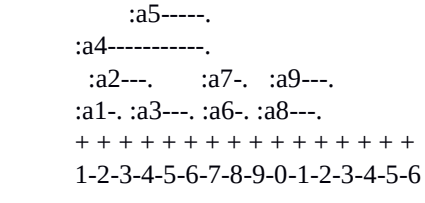
trò chơi	1	2	3	4	5	6	7	8	9
bắt đầu	1	2	4	1	5	8	9	11	13
kết thúc	3	5	7	8	9	10	11	14	16

Có 2 cách lựa chọn được 4 trò là chơi các trò {1,3,6,8} hoặc {2,5,7,9}

Bài tóan này có cần quy hoạch động không? Câu trả lời là không. thuật tóan tham lam ở bài này là thuật tóan hiệu quả.

Tư tưởng tham lam như sau: sau 1 hoạt động , để có thể có nhiều thời gian để chơi những trò kế tiếp, ta chọn những trò chơi có thời gian kết thúc sớm nhất, chơi trò chơi đó và chọn tiếp.Vì lựa chọn trò chơi kết thúc sớm nhất đồng nghĩa việc ta còn nhiều thời gian để tham gia các hoạt động khác. Nếu ta chọn 1 trò chơi khác, thời gian còn lại sẽ ngắn hơn , có thể dẫn tới hậu quả làm được ít việc hơn.

Hãy thử, như ví dụ:



-chơi trò a1, bỏ qua a2,a4;

-chơi trò a3, bỏ qua a5;

-chơi trò a6, bỏ qua a7;

-chơi trò a8, bỏ qua a9;

4/Bài tóan lập lịch trên 2 máy(thuật tóan johnson)

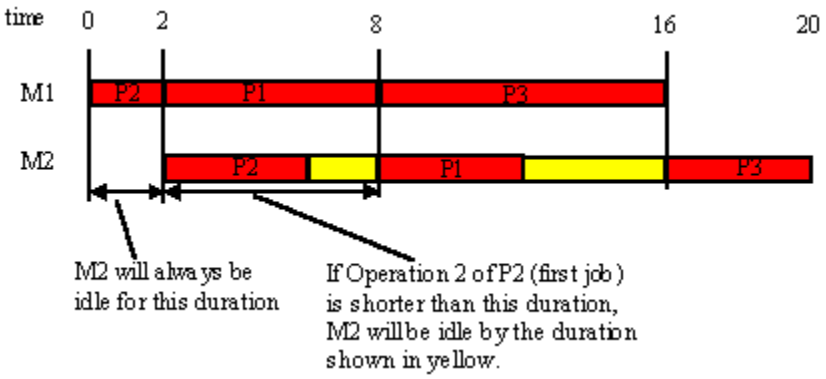
Bài tóan như sau:

Mỗi một chi tiết $D_1,D_2,...D_n$ cần phải được lần lượt gia công trên 2 máy A,B. Thời gian gia công chi tiết D_i trên máy A là a_i trên máy B là b_i (i=1,2..n). Hãy tìm lịch (trình tự gia công) các chi tiết trên hai máy sao cho việc hoàn thành gia công tất cả các chi tiết là sớm nhất.

Ví dụ ta có 3 chi tiết D1,D2,D3 thời gian thực hiện lần lượt như bảng sau:

	D1	D2	D3
máy A	6	2	8
máy B	4	4	4

Ta có thể chọn 1 trong các thứ tự 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, và 3-2-1.Trong mọi trường hợp, máy A luôn hoạt động trong thời gian từ T=0 đến T=6+2+8=16.Còn máy B: nó không thể gia công 1 bộ phận nào nếu máy A chưa gia công bộ phận đó trước. Nếu 1 bộ phận nào đó gia công quá ngắn ở máy B và khi máy B làm xong việc của nó, máy A vẫn chưa đưa được bộ phận tiếp theo vào, nó sẽ làm máy B có 1 khoảng thời gian "vô công rỗi nghề".



Dễ thấy rằng, ta luôn nên để bộ phận cần ít thời gian gia công ở máy A nhất lên đầu, như thế sẽ giảm thời gian chờ của máy B.Cũng nhưng thế, ta nên đặt bộ phận cần ít thời gian gia công ở máy B nhất xuống cuối, nhưng thế sẽ giảm được thời gian đợi từ khi máy A dừng đến khi công việc kết thúc. Còn nhưng công việc ở giữa? Ta sẽ cố gắng giảm đoạn màu vàng (ở hình vẽ) xuống mức tối thiểu. Để đạt được điều này, ta sử dụng thuật tóan johson:

-Đưa những chi tiết cần gia công ngắn nhất ở máy A vào càng sớm càng tốt.

-Cùng lúc đó, đưa những chi tiết gia công lâu nhất ở máy B vào càng sớm càng tốt.

Thuật tóan johnson cài đặt như sau:

Chia các chi tiết thành 2 nhóm: nhóm N1, gồm các chi tiết D_i thoả mãn $a_1 < b_1$, tức là $\min(a_i,b_i) = a_i$ và nhóm N2 gồm các chi tiết D_i thoả mãn $a_i > b_i$ tức là $\min(a_i,b_i)=b_i$. Các chi tiết D_i thoả mãn $a_i = b_i$ xếp vào nhóm nào cũng được.

Sắp xếp các chi tiết trong N1 theo chiều tăng của các a_i và sắp xếp các chi tiết trong N2 theo chiều giảm của các b_i

Nối N2 vào đuôi N1, dãy thu được (đọc từ trái sang phải) sẽ là lịch gia công.

5/Bài toán tìm cây bao trùm nhỏ nhất trên đồ thị vô hướng:

Với một đồ thị liên thông, vô hướng cho trước, cây bao trùm của nó là một đồ thị con có dạng cây và có tất cả các đỉnh liên thông với nhau. Một đồ thị có thể có nhiều cây bao phủ khác nhau. Chúng ta cũng có thể gán một *trọng số* cho mỗi cạnh, là con số biểu thị sự "không ưa thích" và dùng nó để tính toán trọng số của một cây bao trùm bằng cách cộng tất cả trọng số của cạnh trong cây bao trùm đó. Khi đó, một **cây bao trùm nhỏ nhất** là một cây bao trùm có trọng số bé hơn bằng trọng số của tất cả các cây bao trùm khác.

Tổng quát hơn, bất kỳ một đồ thị vô hướng (không nhất thiết liên thông) đều có một **rừng bao phủ nhỏ nhất**, là hội của các cây bao trùm nhỏ nhất của các thành phần liên thông của nó.

Bài toán này có thể dễ dàng giải quyết bằng những thuật toán như Kruskal hay Prim, cả 2 đều là những thuật toán tham lam. Ở bài viết này, tác giả xin nói về thuật toán Kruskal.

Thuật toán Kruskal dựa trên mô hình xây dựng cây khung nhỏ nhất bằng thuật toán hợp nhất :

-Thuật toán không xét các cạnh với thứ tự tuỳ ý.

-Thuật toán xét các cạnh theo thứ tự đã sắp xếp theo trọng số.

Để xây dựng tập n-1 cạnh của cây khung nhỏ nhất - tạm gọi là tập K, Kruskal đề nghị cách kết nạp lần lượt các cạnh vào tập đó theo nguyên tắc như sau:

-Ưu tiên các cạnh có trọng số nhỏ hơn.

-Kết nạp cạnh khi nó không tạo chu trình với tập cạnh đã kết nạp trước đó.

Đó là một nguyên tắc chính xác và đúng đắn, đảm bảo tập K nếu thu đủ n - 1 cạnh sẽ là cây khung nhỏ nhất.

Cách cài thuật tóan Krustal:

Bước 1: Sắp xếp các cạnh của đồ thị theo thứ tự trọng số tăng dần.
Bước 2: Khởi tạo T := Ø
Bước 3: Lần lượt lấy từng cạnh thuộc danh sách đã sắp xếp. Nếu T+{e} không chứa chu trình thì gán T:=T+{e}.
Bước 4: Nếu T đủ n - 1 phần tử thì dừng, ngược lại làm tiếp bước 2.

III/Bài tham khảo về thuật tóan tham lam

Phần khó nhất của việc áp dụng một thuật toán tham lam chỉ đơn giản là nhận ra rằng một vấn đề là tham lam. Lỗi phổ biến nhất là áp dụng tham lam không đúng chỗ. Nhiều vấn đề làm ra để được tham lam, nhưng thực sự đòi hỏi một giải pháp quy hoạch động. Nếu bạn có thể chứng minh một giải pháp tham lam, hãy sử dụng nó, nhưng hãy cẩn thận khi cho rằng giải pháp thực sự là tham lam.

1/DNA khác biệt

(AMPPZ 2012)

Cho một chuỗi DNA độ dài N chỉ gồm các kí tự A, C, G và T. Bạn cần xác định một chuỗi DNA khác với độ dài N sao cho chuỗi mới khác chuỗi DNA ban đầu càng nhiều càng tốt. Định nghĩa chặt chẽ: cho một chuỗi DNA U độ dài N, bạn cần tìm một chuỗi DNA V độ dài N sao cho độ dài dãy con chung dài nhất của U và V là nhỏ nhất.

Ví dụ, với N = 4 và U = GACT, bạn có thể chọn V = TCAG, đạt được dãy con chung dài nhất chỉ chứa đúng 1 kí tự

Giới hạn: N <= 10000

Giải thuật: chọn kí tự nào ít xuất hiện nhất ở chuỗi DNA đầu thì in ra chuỗi DNA gồm tòan kí tự đó.

Chứng minh:

Ta sẽ chứng minh rằng với mọi xâu DNA V độ dài N thì xâu con chung dài nhất của U cho trước với V sẽ không nhỏ hơn số lần xuất hiện của kí tự xuất hiện ít nhất. Tức là nếu gọi số lần kí tự c xuất hiện trong xâu s là t(c, s) thì $|LCS(U,V)| \geq \min\{t(A,U), t(C,U), t(G,U), t(X,U)\} = X$.

Giả sử $|LCS(U,V)| < X$ thì với mọi kí tự $c \in \{A,C,G,T\}$ ta sẽ có $t(c,V) < t(c,U)$ vì nếu $t(c,V) \geq t(c,U)$ thì $|LCS(U,V)| \geq t(c,U) \geq X$. Từ đó suy ra $|V| < |U|$, vô lí vì $|U|=|V|=N$.
Vậy để độ dài xâu con chung giữa U và V là nhỏ nhất thì xâu V sẽ gồm N kí tự là kí tự xuất hiện ít nhất trong U . Bài toán có lời giải O (N).

2/Barn Repair

(1999 USACO Spring Open)

Với M ($1 \leq M \leq 50$) là số lượng tối đa các tấm lợp chuồng bò có thể mua, mỗi tấm lợp có thể phủ một nhóm liên tiếp các chuồng bò có độ dài bất kỳ. S là số lượng chuồng bò ($1 \leq S \leq 50$). Và C ($1 \leq C \leq S$) con bò, mỗi con bò nằm trong một chuồng bò. Hãy tính số lượng chuồng bò tối thiểu phải phù để tất cả các chuồng có bò đều phải được phủ.

INPUT

Dòng 1 là ba số M, S, và C. C dòng tiếp theo, mỗi dòng ghi 1 số A[i] là số hiệu chuồng chứa bò
OUTPUT
Số chuồng bò tối thiểu phải phủ.

Sample Input	Sample Output
4 50 18 3 4 6 8 14 15 16 17 21 25 26 27 30 31 40 41 42 43	25

Giải thuật:

Đầu tiên, ta dễ dàng đưa ra giải pháp quy hoạt động : F[i][j] là số chuồng cần phủ tối thiểu để phủ kín i chuồng có bò đầu tiên với j tấm lợp, với mỗi F[i][j] ta tìm 1 số F[u][j-1] (0<u<i) để tối ưu hóa cho F[i][j], độ phức tạp là O(M*C^2) , không quá lớn với máy tính hiện đại ngày nay, nhưng hãy nhớ đề bài ra vào năm 1999, hơn nữa, chúng ta không việc gì phải làm một bài với độ phức tạp lớn như vậy cả khi có thể làm với độ phức tạp nhỏ hơn rất nhiều sử dụng tư tưởng tham lam:

Ta có nhận xét: với M>=C , đáp án chính là C. Với M<C, ta giả sử ta có C tấm lợp để phủ cho C chuồng bò, số chuồng bò cần phủ là C chuồng, vì C>M, ta cố gắng kéo dài các tấm lợp của 1 số cặp chuồng có bò cạnh nhau (tức là ở giữa chúng không có chuồng nào có bò) để chúng nối với nhau tạo thành 1 tấm lợp duy nhất, mỗi lần ta nối 2 tấm lợp thành 1, số tấm lợp cần dùng sẽ giảm đi 1, số lần "kéo dài và nối" là C-M lần, tư tưởng tham lam ở đây là ta chỉ chọn C-M cặp chuồng có bò sao cho khoảng cách giữa chúng là nhỏ nhất có thể vì như thế số chuồng lợp thừa cũng là nhỏ nhất. Với tư tưởng này, đầu tiên ta cần sắp xếp các số A[i] theo thứ tự, sau đó, ta cần sắp xếp các khoảng cách của các cặp chuồng có bò cạnh nhau rồi chọn ra C-M cặp chuồng bò, độ phức tạp chung là O(C*log(C)) - nhỏ hơn nhiều độ phức tạp QHĐ ở trên.

3/*Milk Queue*

(US OPEN, April 2006 USACO Contest)

Mỗi sáng,N(1<=N<=25000) con bò của nông dân John xếp hàng đi lấy sữa.Với nỗ lực sắp xếp qúa trình vắt sữa, ông John đưa ra quy trình gồm 2 giai đoạn mà đàn bò sẽ lần lượt đi qua 2 khu

vắt sữa bò, và mỗi con bò sẽ được vắt sữa ở cả 2 khu.Ông John tự tay vắt sữa từng con bò ở khu 1, còn bạn thân của ông: nông dân Rob thì vắt sữa ở khu 2. Những con bò đi ra khỏi khu 1 sẽ tiến vào khu 2.

Không may là đàn bò của ông John lại đi vào 2 khu theo thứ tự không hiệu quả.Ví dụ, nếu nông dân John mất quá lâu để vắt sữa những con bò đặc biệt, nông dân Rob có thể ngồi nhàn rỗi không có gì để làm trong một thời gian. Mặt khác, nếu nông dân John làm việc quá nhanh sau đó có thể kết thúc với một danh sách dài các con bò chờ đợi để vào khu thứ hai.

Hãy giúp ông John tìm ra cách sắp xếp thứ tự để lấy sữa xong càng sớm càng tốt.

Input

Dòng 1: chứa số nguyên N

Dòng 2.. N+1: dòng i+1 chứa 2 số nguyên A[i],B[i] là thời gian lấy sữa của con bò thứ i ở khu 1 và khu 2.

Output

Thời gian lấy sữa khi được sắp xếp tối ưu.

Sample Input	**Sample Output**
3 2 2 7 4 3 5	16

Nếu coi 2 khu vắt sữa là 2 máy, mỗi con bò là 1 chi tiết cần gia công, thì bài tóan này chính là bài tóan lập lịch trên 2 máy. Ta có thể dễ dàng sử dụng thuật toán Johnson giải quyết bài tóan này.

4/ Backup

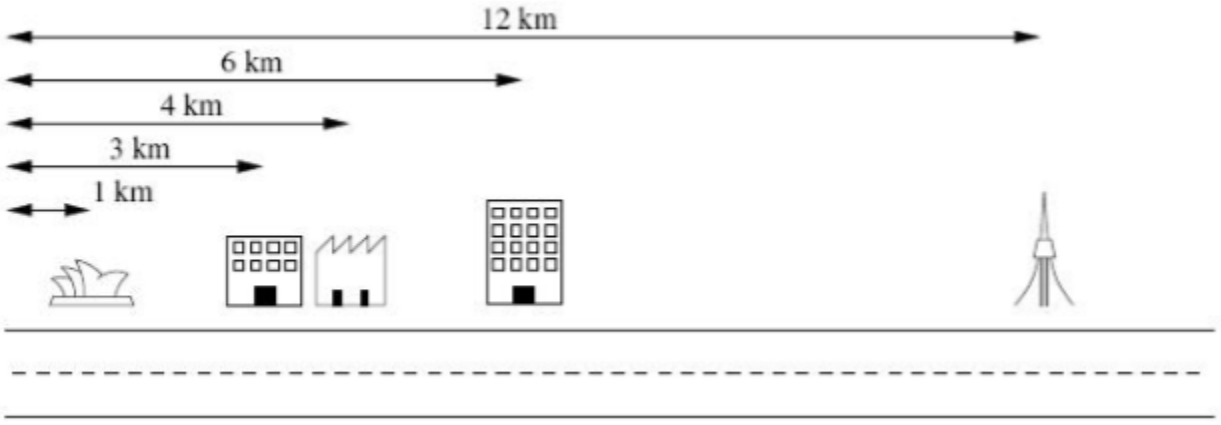
(APIO 2007)

Bạn điều hành một công ty công nghệ chuyên sao lưu dữ liệu cho những văn phòng lớn. Sao lưu dữ liệu không phải là việc gì vui vẻ cả, bạn cần phải thiết kế hệ thống sao cho các văn phòng có thể sao lưu dữ liệu lẫn nhau trong lúc bạn ngồi nhà và uống trà đá.

Các văn phòng này đều nằm trên cùng một tuyến phố. Khi bạn nối 2 văn phòng lại với nhau, bạn nối một mạng cáp giữa 2 tòa nhà, và như thế, chúng có thể sao lưu dữ liệu lẫn nhau.

Tuy nhiên, dây cáp dữ liệu rất đắt và công ti chỉ cho bạn tối đa k dây cáp. Tức là bạn có thể nối k cặp văn phòng với nhau (tổng cộng 2k văn phòng).Không văn phòng nào được nối cùng lúc với 2 văn phòng khác (bạn phải chọn 2k văn phòng khác nhau).Giá cáp tăng theo cấp số nhân

mỗi km, bạn phải chọn k cặp văn phòng sao cho tổng khoảng cách giữa các cặp nối với nhau là nhỏ nhất (tổng độ dài dây cáp sử dụng là nhỏ nhất).



Ví dụ, bạn có 5 văn phòng ở km số 1,3,4,6,12 và k=2. Cách tốt nhất là chọn nối văn phòng 1 với 2 , 3 với 4. tổng khỏang cách sẽ là 3-1+6-4=4

INPUT

Dòng đầu chứa 2 số nguyên n và k là số văn phòng và số dây cáp được dùng (2<=n<=100000; 1<= k<= n/2).

n dòng tiếp theo mỗi dòng chứ 1 số nguyên s (0 <= s <= 1 000 000 000), thể hiện khỏang cách giữa mỗi văn phòng và đầu phố.Chúng được sắp xếp theo thứ tự không giảm.

Không có 2 văn phòng nào cùng địa điểm.

OUTPUT

Tổng độ dài nhỏ nhất để nối 2k văn phòng thành k cặp.

Sample Input	Sample Output
5 2 1 3 4 6 12	4

Đầu tiên, ta nghĩ ngay đến thuật tóan quy hoạch động: gọi F[i][j] là cách tốt nhất nối j cặp văn phòng trong i văn phòng đầu tiên. Ta có 2 lựa chọn: nối toà nhà i vào toàn nhà i-1 hoặc không làm gì cả. F[i][j]=min(F[i-1][j],F[i-2][j-1]+dist[i]-dist[i-1]) . Với độ phức tạp O(N*K) bạn khó có thể qua được tất cả các test.

Có một cách tham lam đơn giản đó là chọn ra 2 văn phòng cạnh nhau có khoảng cách nhỏ nhất. Thêm cạnh nối giữa 2 văn phòng đó vào tập chọn, xóa đi tất cả các cạnh nối với 2 văn phòng đó. Lặp lại đến khi chọn đủ k cặp. Tuy nhiên, đây là cách tham lam sai, ví dụ như:

A <-11-> B <-5-> C <-3-> D <-6->E <-14-> F <-10-> G <-9-> H <-10-> I

với k=4. Với cách tham lam trên, đầu tiên ta chọn cạnh CD,bỏ đi BC,DE , ta chọn cạnh GH,bỏ đi FG và HI chọn tiếp AB và EF. Tổng độ dài là 3+9+11+14=37.

Tuy nhiên, nếu chọn {BC=5, DE=6, FG=10, HI=10} tổng độ dài chỉ có 31.

Vì vậy , ta phải cải tiến phương pháp tham lam này: Khi ta lấy cạnh CD vào tập kết quả, ta tạo 1 cạnh ảo giữa B và E với BE=BC+DE-CD nghĩa là khi thêm BE vào tập kết quả, ta bỏ đi cạnh CD và thêm cặp cạnh BC và DE vào. Điều này đảm bảo số lượng cạnh lấy ra vẫn là k.

Ví dụ ở trên:

Bước 1: ta lấy ra cạnh CD, ans=CD=3, tạo cạnh ảo BE=BC+DE-CD=8.

A <-11-> B C **3** D E <-14-> F <-10-> G <-9-> H <-10-> I

|_____|
8

Bước 2: ta thấy cạnh BE là cạnh ngắn nhất, ta chọn BE, ans=ans+BE=11. Tạo cạnh ảo AF=AB+EF-BE=17.

A <-11-> B C **3** D E <-14-> F <-10-> G <-9-> H <-10-> I

| * * |
| ******** 8 ******* |
----------17----------

Bước 3: Cạnh GH là ngắn nhất, chọn GH, ans=ans+GH=20. Tạo cạnh ảo FI=FG+HI-GH=11.

A <-11-> B C **3** D E <-14-> F <-10-> G **9** H <-10-> I

| * * | |
| ******** 8 ******* | |
----------17---------- ----------11----------

Bước 4: Thấy cạnh FI nhỏ nhất, chọn cạnh FI, ans=ans+FI=31. Đã chọn đủ k cạnh, kết thúc vòng lặp, trả về kết quả ans=31.

Khi ta đưa 1 cạnh vào tập hợp các cạnh được chọn, ta duy trì khả năng xóa cạnh đó ở trong cạnh ảo được tạo. Khi muốn xóa 1 cạnh để được kết quả tốt hơn, đòi hỏi phải thêm cả 2 cạnh liền kề cạnh đó vào tập cạnh kết quả.

Ta duy trì 1 cái heap các cạnh, khi ta lấy ra 1 cạnh (i,j) có độ dài nhỏ nhất, ta đưa cạnh ảo (i-1,j+1) vào trong heap. Ta tốn O(N) để xây dựng heap và k lần update heap mỗi lần mất O(logN) . Tổng cộng độ phức tạp là O(N+KlogN), nhỏ hơn nhiều lần độ phức tạp của phương pháp QHĐ và có thể dễ dàng qua được hết test.

BÀI LUYỆN TẬP: SPOJ:CBUYING, MJOURNEY,TWO,NOIXICH,QBMST,CINEMA.