

## EulerTree

Un robot esegue una visita DFS in un grafo di  $n$  nodi e  $m$  archi, raggiungendone tutti i nodi, e registra la storia della sua visita nel seguente file

```
input.txt
5 8
0 1
1 2
2 0
2 3
3 0
3 1
1 4
4 0
```

In questo caso il grafo aveva  $n=5$  nodi e  $m=8$  archi. In una visita DFS ogni arco viene attraversato precisamente due volte: una in ogni direzione. Tuttavia, man mano che la visita procede, ad ogni attraversamento di arco, il robot riporta tale arco nella lista del file input.txt solamente se e' la prima volta che l'arco viene attraversato. Ciascun arco viene pertanto riportato in input.txt al momento del suo primo attraversamento, e cosi' gli archi compaiono altresì ordinati per tempo crescente di primo attraversamento. Dalla storia riportata in input.txt deduciamo che la posizione del robot nei vari istanti era come da tabella:

istante:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
nodo:	0	1	2	0	2	3	0	3	1	3	2	1	4	0	4	1	0	

$$\begin{array}{c} 4 \\ / \quad \backslash \\ 0 \text{---} 1 \\ \backslash \quad / \\ | \quad | \\ | \quad | \\ / \quad \backslash \\ 3 \text{---} 2 \end{array}$$

Gli archi  $(0,1)$ ,  $(1,2)$ ,  $(2,3)$  e  $(1,4)$ , che portano alla scoperta di un nuovo nodo, vengono chiamati tree-edges in quanto costituiscono un'arborecenza che partendo dal nodo radice (nodo 0) fornisce un unico percorso per raggiungere ogni altro nodo.

Ora che il grafo e' stato rilevato con la visita, siamo chiamati a calcolare il minimo costo di un percorso chiuso (nodo di partenza = nodo di arrivo) che attraversi ogni arco (non importa la direzione) almeno una volta.

In seno al percorso, ogni attraversamento di un arco costa 1 se esso era stato labelato come un tree-edge secondo la visita riportata in input.txt e costa  $m$  altrimenti. Il costo del percorso e' dato dalla somma dei costi degli attraversamenti che esso prescrive.

Risolviamo correttamente l'istanza input.txt sopra discussa se scriviamo su directory corrente il file:

```
output.txt
37
```

in quanto esiste un percorso chiuso che attraversa 2 volte il tree-edge  $(2,3)$  e precisamente una volta ogni altro arco.

Assunzioni:

- tempo limite = 1 secondo (user time);
- il grafo puo' contenere piu' archi con gli stessi estremi (archi paralleli) ma non contiene loops (un arco i cui due estremi coincidono);
- i nodi sono numerati da 0 ad  $n-1$ , e 0 e' il nodo di partenza del robot;
- $n \leq 100$  e  $m \leq 1000$ ;
- in almeno 3 istanze il grafo e' un albero;
- in almeno 10 istanze  $m \leq 20$ ;
- per ogni arco listato in input.txt, l'ordine degli estremi rispetta il verso

## EulerTree

di percorrenza dell'arco da parte del robot (al primo attraversamento);

- i tree-edges (ossia quegli archi della lista per i quali il secondo estremo compare per la prima volta nella lista) formano un'alborescenza radicata in 0 che raggiunge tutti gli  $n$  nodi;
- non e' necessaria per risolvere agevolmente il problema, ma l'assunzione che il robot ha seguito una logica DFS puo' essere sfruttata. La logica DFS impone al robot di ripercorrere immediatamente a ritroso l'arco appena percorso se esso lo ha condotto in un nodo gia' visto (e l'arco era stato percorso solo una volta), altrimenti, se il robot e' in un nodo nuovo oppure non puo' ripercorrere a ritroso l'ultimo arco (in quanto gia' percorso in quella direzione), allora potra' ripercorrere quell'arco che lo ha condotto per la prima volta nel nodo attuale solo dopo che ogni altro arco facente capo al nodo e' stato percorso.

Nota: vi lascio a disposizione il generatore di istanze che utilizzerò io, con makefile e shell-script makeInputs.sh (ovviamente cambierò i seeds al momento della valutazione).