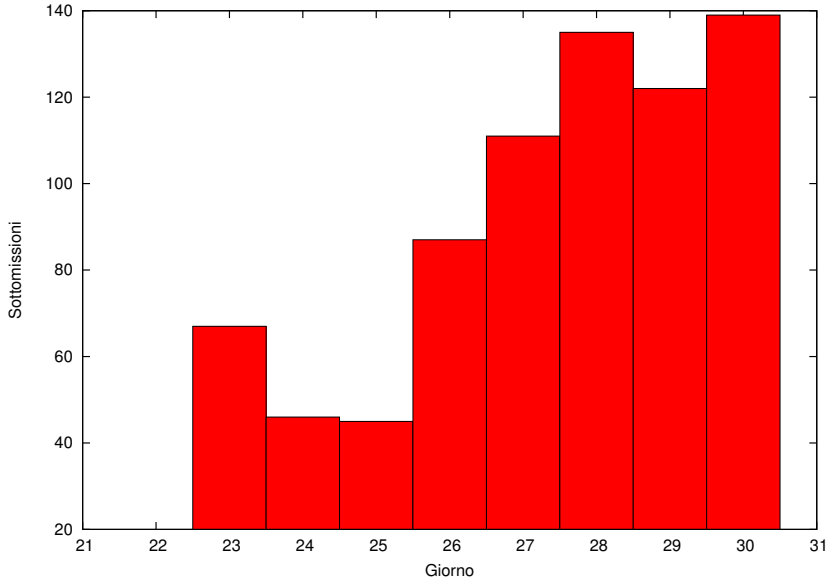
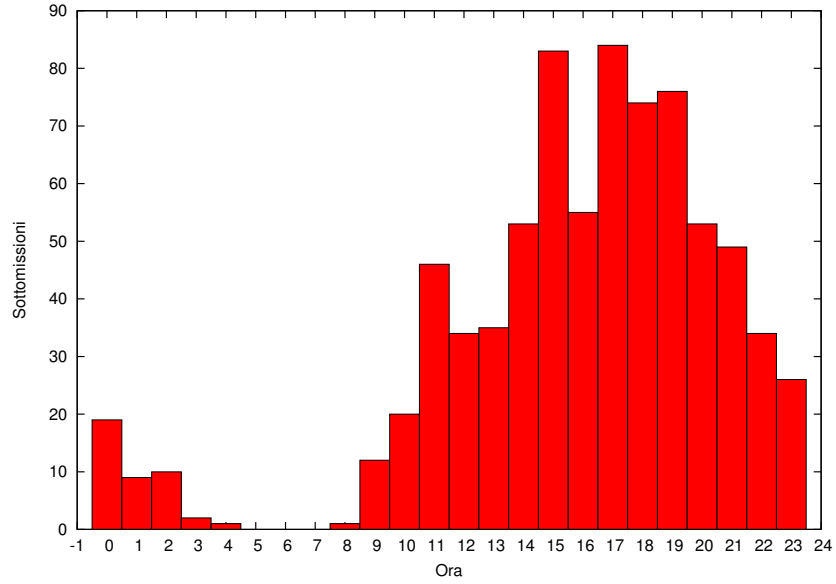
Spiegazioni ASD 2015
Mondodisco

Risultati

Statistiche

Numero sottoposizioni: 726





Punteggi

- $P < 30 \rightarrow$ progetto non passato
- $30 \leq P \leq 50 \rightarrow$ un punto bonus al voto dello scritto
- $55 \leq P \rightarrow$ due punti bonus al voto dello scritto
- Classifica completa sul mio sito

Soluzione base: K=0

Definizione alternativa

Per ogni nodo entrata, calcolare la sua distanza dalla destinazione.

Approcci

Visita in ampiezza (erdos) per grafo non pesato, Dijkstra, Johnson o Bellman-ford per grafo pesato

Ottimizzazioni

1. Invece che eseguire l'algoritmo da ogni entrata, eseguirlo un unica volta sul grafo trasposto a partire dalla destinazione
2. Fino a questo punto i grafi sono aciclici: invece di usare strutture dati complesse, basta fare ordinamento topologico e visitare i nodi in ordine.

K=1

Idee

Supponiamo di essere in un certo nodo. Le opzioni sono due: (1) i cattivi utilizzeranno il loro potere, (2) gli eroi saranno liberi di scegliere l'arco che gli aggrada.

1. Se utilizzano il potere, i cattivi manderanno gli eroi sempre nel nodo con distanza massima dalla destinazione
2. Se non utilizzano il potere, i buoni andaranno nel nodo che minimizza la distanza *considerando che i nemici hanno ancora un potere*

K=1: definizioni

Dato nodo n definiamo:

1. d_n la distanza (pesata) da n alla destinazione
2. $Good_n$ quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici non usano il loro potere in n
3. Bad_n quanto impiegherannno gli eroi a raggiungere la destinazione da n se i nemici usano il loro potere in n
4. Sol_n quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici hanno un potere da usare in maniera ottimale

Note:

1. d_n l'abbiamo già calcolato per risolvere il caso $K = 0$
2. Sol_n è il valore che dobbiamo calcolare come output se n è un entrata

K=1: definizioni

Dato nodo n definiamo:

1. d_n la distanza (pesata) da n alla destinazione
2. $Good_n$ quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici non usano il loro potere in n
3. Bad_n quanto impiegherannno gli eroi a raggiungere la destinazione da n se i nemici usano il loro potere in n
4. Sol_n quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici hanno un potere da usare in maniera ottimale

Per calcolare Bad_n, sappiamo che se i nemici usano il loro potere ci manderanno lungo la strada più lunga. Visto che non avranno più poteri, possiamo utilizzare direttamente la distanza.

$$Bad_n = max_{v \in vic(n)}(d_v + w((n, v)))$$

K=1: definizioni

Dato nodo n definiamo:

1. d_n la distanza (pesata) da n alla destinazione
2. $Good_n$ quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici non usano il loro potere in n
3. Bad_n quanto impiegherannno gli eroi a raggiungere la destinazione da n se i nemici usano il loro potere in n
4. Sol_n quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici hanno un potere da usare in maniera ottimale

Gli eroi, se lasciati liberi, andranno per la strada migliore considerando che i nemici possono ancora utilizzare il potere

$$Good_n = min_{v \in vic(n)}(Sol_v + w((n, v)))$$

K=1: definizioni

Dato nodo n definiamo:

1. d_n la distanza (pesata) da n alla destinazione
2. $Good_n$ quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici non usano il loro potere in n
3. Bad_n quanto impiegherannno gli eroi a raggiungere la destinazione da n se i nemici usano il loro potere in n
4. Sol_n quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici hanno un potere da usare in maniera ottimale

La soluzione richiede solo la combinazione delle due proprietà: i nemici decideranno se utilizzare il loro potere o no a seconda di quale delle due scelte massimizza il tempo necessario agli eroi per raggiungere la destinazione.

$$Sol_n = max(Good_n, Bad_n)$$

Soluzione

Per calcolare d_n, Bad_n, $Good_n$ e Sol_n bisogna aver già calcolato questi valori per i vicini di n

Soluzione

Eseguiamo l'ordinamento topologico sul grafo e processiamo i nodi in ordine inverso (partendo dai pozzi), calcolando per ogni nodo i valori.
Questo processo ci calcolerà l'output da ogni possibile posizione di entrata.
Complessità: $O(N + E)$

Casi con K generico

Ripetiamo l'approccio per ogni K.
Dato nodo n definiamo:

1. $Sol_{n,k}$ quanto impiegheranno gli eroi a raggiungere la destinazione da n se i nemici hanno k poteri da usare in maniera ottimale
2. $Good_{n,k}$ quanto impiegheranno gli eroi a raggiungere la destinazione da n (con i nemici che hanno k poteri da usare) se i nemici non usano il loro potere in n gli rimangono k
3. $Bad_{n,k}$ quanto impiegherannno gli eroi a raggiungere la destinazione da n (con i nemici che hanno k poteri da usare) se i nemici usano il loro potere in n

$$Bad_n = max_{v \in vic(n)}(Sol_{v,k-1} + w((n,v)))$$

$$Good_n = min_{v \in vic(n)}(Sol_{v,k} + w((n,v)))$$

Grafo con cicli

Se il grafo non è aciclico, non possiamo usare l'ordinamento topologico per indicare l'ordine in cui visitiamo i nodi.
L'approccio richiesto richiede l'utilizzo di dijkstra che, per ogni valore di k, visita i nodi utilizzando una coda a priorità che indicizza i nodi in ordine crescente di $Sol_{n,k}$

Note

- Classifiche e sorgenti sul mio sito (controllate numeri di matricola)
- Chi ha passato il primo progetto non è obbligato a fare il secondo
- Assumo gli stessi gruppi, in caso di cambiamenti scrivetemi.