

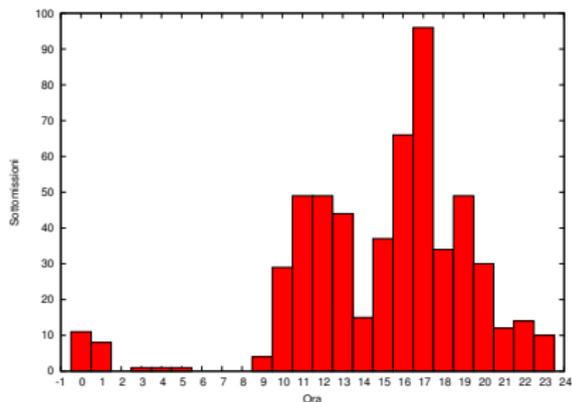
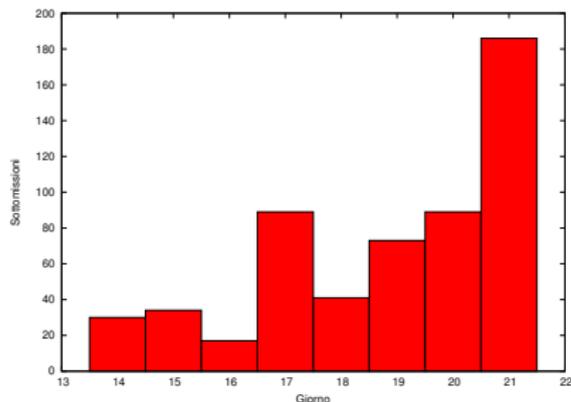
Spiegazioni ASD 2014

Batman si annoia

Risultati

Statistiche

Numero sottoposizioni: 534



Punteggi

- ▶ $P < 30 \rightarrow$ progetto non passato
- ▶ $30 \leq P \leq 60 \rightarrow$ un punto bonus al voto dello scritto
- ▶ $65 \leq P \rightarrow$ due punti bonus al voto dello scritto
- ▶ Classifica completa sul mio sito
- ▶ Controllo per copiatore in corso

Posizione di nemici

Definizione

Un arco $x \rightarrow y$ contiene un nemico se non esiste un percorso fra y e x .

Definizione equivalente

Un arco $x \rightarrow y$ contiene un nemico se x e y fanno parte di una diversa componente fortemente connessa.

Grafo aciclico (30 punti)

Grafo aciclico \rightarrow Ogni nodo forma una componente fortemente connessa

Basta contare il numero di percorsi fra sorgente e destinazione

```
void visit(nodo n, int& cammini){  
    if (n==DEST)  
        cammini++;  
    else  
        for (vicino:n.vicini)  
            visit(vicino, cammini);  
}
```

Nota: complessità dipende dal numero di percorsi

Grafo aciclico (50 punti)

Calcoliamo il numero di percorsi dalla sorgente ad ogni altro nodo:

Il numero di percorsi di un nodo v è la somma del numero di percorsi dei predecessori

$$NP(v) = \sum_{(w,v) \in E} NP(w)$$

Basta passare i nodi secondo l'ordinamento topologico:

```
Sorgente.numpercorsi=1
for (nodo:G.topOrder){
    for (vicino:nodo.vicini)
        vicino.numpercorsi+=nodo.numpercorsi
}
```

Caso generale

Dato un grafo qualunque, i nemici si trovano su archi fra componenti fortemente connesse:

Calcolare componenti connesse

Kosaraju o Tarjan

Calcolare grafo delle componenti

Un nodo per ogni componente, Un arco per ogni nemico.

Riutilizzare algoritmo per grafo aciclico

Il grafo delle componenti è aciclico.

Note

- ▶ Tarjan e Kosaraju trovano le componenti già in ordine topologico
- ▶ Ottimizzazioni che riducono il grafo possono aiutare, ma non sono necessarie
- ▶ Il calcolo dei percorsi può essere risolto via memoization
- ▶ Classifiche sul mio sito.
- ▶ Chi ha passato il primo progetto non è obbligato a fare il secondo