
Cognome: Nome: Matricola:

Ogni risposta deve essere adeguatamente motivata.

Domanda 1

Trovare tutti gli alberi di grado minimo 2 che memorizzano l'insieme di chiavi $\{1, 2, 3, 4, 5\}$.

Domanda 2

Si consideri la struttura dati per insiemi disgiunti basata su foreste discussa a lezione, con unione per rango e compressione dei cammini. Si dimostri che, se m è il numero di elementi memorizzati nella struttura (operazioni *MakeSet* eseguite), allora, per ogni nodo x vale $\text{rank}(x) \leq \log_2(m)$.

Domanda 3

Si consideri ancora la struttura dati per insiemi disgiunti basata su foreste discussa a lezione, con unione per rango e compressione dei cammini. Si valuti la complessità di una operazione $findSet(x)$ in funzione del numero m di elementi contenuti nella struttura.

(Sugg.: Dimostrare che per ogni nodo x , l'altezza dell'albero a cui x appartiene è limitata da $rank(x)$.)

Cognome: Nome: Matricola:

Esercizio 1

Progettare una struttura dati per la gestione di un insieme di interi, con operazioni

- $New(S)$ crea un insieme vuoto;
- $Ins(S, x)$ inserisce l'elemento x nell'insieme S ;
- $Cut(S, n)$ cancella da S tutti gli elementi maggiori di n .

Si richiede che una qualsiasi sequenza di n operazioni venga eseguita in tempo $O(n \log n)$.

- a. Specificare quali strutture dati di supporto si utilizzano e fornire lo pseudo-codice per la realizzazione delle operazioni suddette.
- b. Dimostrare, mediante un'analisi ammortizzata della complessità basata sul metodo del potenziale o degli accantonamenti, che una sequenza di n operazioni costa $O(n \log n)$.



Cognome: Nome: Matricola:

Esercizio 2

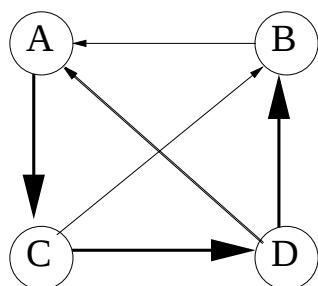
Supponiamo di avere N (N dispari) votanti e M candidati. Consideriamo il seguente grafo orientato (detto grafo di maggioranza):

- nodi: uno per ogni candidato
- archi: un arco dal nodo del candidato A al nodo del candidato B se la maggioranza dei votanti preferisce A a B .

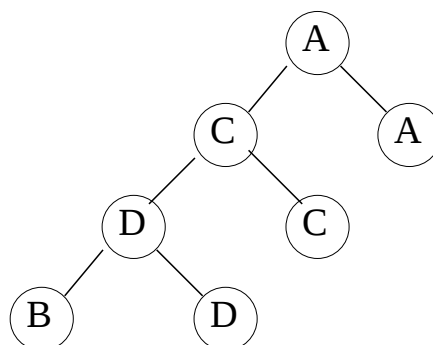
Supponiamo ora di scegliere il vincitore facendo un torneo dove i candidati si confrontano a coppie e la vittoria è determinata dalle informazioni del grafo di maggioranza. Un torneo si può rappresentare come un albero binario con M foglie (una per ogni candidato) e dove ogni nodo interno è etichettato con il vincitore tra i suoi figli. Nella radice ci sarà quindi il vincitore del torneo.

Si può dimostrare che, dato un grafo di maggioranza, è possibile costruire un torneo (i.e., albero binario) in modo tale che un determinato candidato X vinca se e solo se nel grafo di maggioranza esiste un cammino da X a tutti gli altri nodi.

In Figura è mostrato un esempio con 4 candidati.



Grafo di maggioranza
Cammino da A agli altri
evidenziato



Torneo in cui A vince

Scrivere un algoritmo che, dati in input un grafo di maggioranza e un candidato X , determini se esiste un torneo in cui X può vincere.

Cognome: Nome: Matricola:

Domanda 4

Date le ipotesi dell'Esercizio 2, supponiamo che un candidato X sia considerato peggiore di tutti gli altri candidati dalla maggioranza dei votanti.

- Cosa succede nel grafo di maggioranza?
- Dimostrare che non esiste nessun torneo in cui X vince.

