03AAX Algoritmi e Strutture Dati
03MNO Algoritmi e Programmazione
III Appello del 27/06/2022 - Prova di teoria (12 punti)

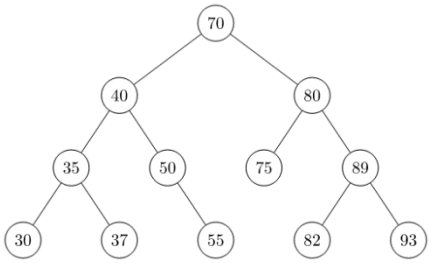
1. (2 punti)
Si determini mediante un algoritmo greedy un codice di Huffman ottimo per i seguenti simboli con le frequenze specificate: B: 5 C: 11 D: 13 F: 12 J: 13 L: 16 R: 8 X: 4 Z: 7. Il ramo sinistro abbia etichetta 0, quello destro etichetta 1. Si elenchi il contenuto della coda a priorità per frequenze crescenti di lettere/aggregati ad ogni passo dell'algoritmo.

2. (2 punti)
Sia data la sequenza di chiavi intere 13 181 267 302 98 110 45 207. Si riporti il contenuto di una tabella di hash di dimensione 13, inizialmente supposta vuota, in cui avvenga l'inserimento della sequenza indicata. Si usi il linear chaining.

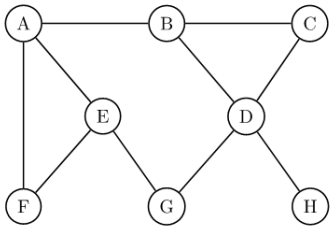
3. (1.5 punti)
Si supponga di aver memorizzato tutti i numeri tra 0 e 1000 in un BST e di cercare la chiave 255. Quali tra le seguenti sequenze non possono essere incontrate durante la ricerca, essendo il BST corretto?

924 220 911 244 898 250 362 255 — 925 202 911 240 912 245 255
2 300 387 219 266 382 381 278 255 — 2 252 401 398 330 344 397 255

4. (1.5 punti)
Si cancelli la chiave 70 dal seguente BST:

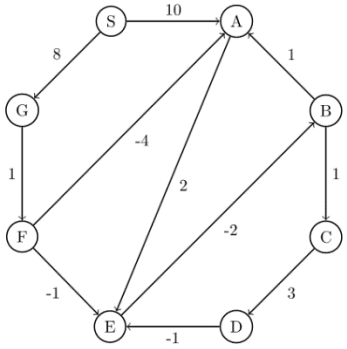


5. (0,5 + 0,5 + 2 punti)
Dato il seguente grafo non orientato



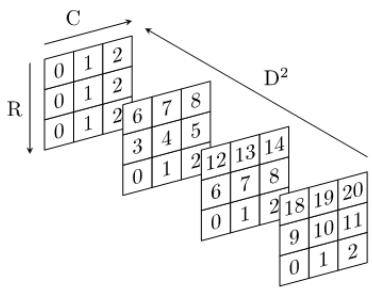
lo si rappresenti come lista e matrice delle adiacenze (**0,5 + 0,5 punti**) e se ne effettui una visita in ampiezza a partire dal vertice **A** (**2 punti**).

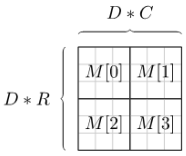
6. (2 punti)
Si determinino per il seguente grafo orientato pesato mediante l'algoritmo di Bellman-Ford i valori di tutti i cammini minimi che collegano il vertice **S** con ogni altro vertice.



03AAX Algoritmi e Strutture Dati
03MNO Algoritmi e Programmazione
III Appello del 27/06/2022 - Prova di programmazione (12 punti)

1. (**2 punti**)
Una funzione riceve una matrice tridimensionale `M` di dimensioni `(D²)xRxC`. La funzione `flatten` alloca una nuova matrice bidimensionale di dimensioni `(DxR)x(DxC)`, ricopia i contenuti della matrice ricevuta come argomento con lo schema illustrato a seguire e ritorna la nuova matrice. In sostanza, nella matrice risultante ogni M[i]-esimo livello viene riposizionato "per righe". Nell'esempio seguente `D` vale 2.





18	19	20	12	13	14
9	10	11	6	7	8
0	1	2	0	1	2
6	7	8	0	1	2
3	4	5	0	1	2
0	1	2	0	1	2

2. (4 punti)
Sia dato un albero di grado `N`, i cui nodi sono definiti dalla seguente struttura C:

```
struct node {
  char *key;
  struct node *children[N];
};
```

Definire la struttura wrapper dell'albero `nTREE` come ADT di prima categoria evidenziando la suddivisione tra file dei vari contenuti. Si realizzi una funzione wrapper `int countIf(nTREE t);` e la corrispondente funzione di visita ricorsiva, che visiti l'albero e ritorni il conteggio del numero di nodi aventi grado (numero di figli) maggiore del grado del rispettivo nodo padre. Il nodo radice, che non ha un padre, conta 1 per default.

3. (6 punti)
Si vogliono generare tutte le sequenze alfabetiche di `k` (parametro del programma) caratteri che rispettino i seguenti vincoli:

- si possono usare solo lettere minuscole (da 'a' a 'z') e maiuscole (da 'A' a 'Z')
- al massimo la metà dei caratteri possono essere lettere minuscole
- lo stesso carattere, senza distinguere tra maiuscolo o minuscolo, non può apparire più di `p` (parametro del programma) volte consecutivamente.

- Si scriva una funzione ricorsiva in C in grado di generare tutte le sequenze secondo le regole di cui sopra, stampando solo quelle accettabili.
- Si giustifichi la scelta del modello combinatorio adottato.
- Si descrivano i criteri di pruning adottato o il motivo della loro assenza.

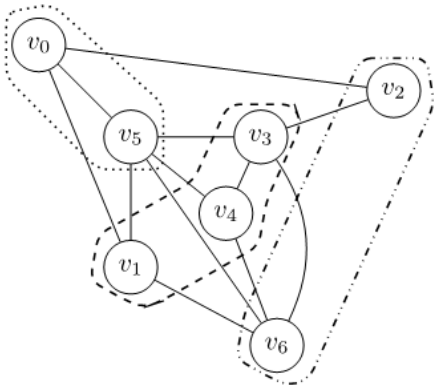
PER ENTRAMBE LE PROVE DI PROGRAMMAZIONE (18 o 12 punti):

- ***indicare nell'elaborato e nella relazione nome, cognome e numero di matricola.***
- ***se non indicato diversamente, è consentito utilizzare chiamate a funzioni standard, quali ordinamento per vettori, funzioni su FIFO, LIFO, liste, BST, tabelle di hash, grafi e altre strutture dati, considerate come librerie esterne.***
- ***gli header file devono essere allegati all'elaborato (il loro contenuto riportato nell'elaborato stesso). Le funzioni richiamate, inoltre, dovranno essere incluse nella versione del programma allegata alla relazione. I modelli delle funzioni ricorsive non sono considerati funzioni standard.***
- ***consegna delle relazioni (per entrambe le tipologie di prova di programmazione): entro giovedì 30/06/2022, alle ore 23:59, mediante caricamento su Portale. Le istruzioni per il caricamento sono pubblicate sul Portale nella sezione Materiale). QUALORA IL CODICE CARICATO CON LA RELAZIONE NON COMPILI CORRETTAMENTE, VERRÀ APPLICATA UNA PENALIZZAZIONE. Si ricorda che la valutazione del compito viene fatta, senza la presenza del candidato, sulla base dell'elaborato svolto in aula. Non verranno corretti i compiti di cui non sarà stata inviata la relazione nei tempi stabiliti.***

03AAX Algoritmi e Strutture Dati
03MNO Algoritmi e Programmazione
III Appello del 27/06/2022 - Prova di programmazione (12 punti)

Dato un grafo G = (V, E) non orientato non pesato, viene definita **domatic partition** una partizione $\{V_1, V_2, \ldots, V_k\}$ dei vertici del grafo tale per cui ogni sottoinsieme V_i è un **dominating set** per il grafo stesso.
Si definisce **dominating set** un sottoinsieme D dei vertici di un grafo tale per cui ogni vertice non in D è adiacente ad almeno un vertice di D.

Esempio:
Sia dato il seguente grafo,



la partizione di V $\{V_1 = \{v_0, v_5\}, V_2 = \{v_1, v_3, v_4\}, V_3 = \{v_2, v_6\}\}$ è una **domatic partition** in quanto soddisfa i criteri per essere una partizione ed ognuno dei sottoinsiemi che vi appartengono è un **dominating set.** La sua cardinalità è 3 ed è anche la cardinalità massima.

Si legga in una opportuna struttura dati un grafo G dal file `g.txt` che ha il seguente formato:
- sulla prima riga compare il numero di vertici V
- segue un numero non definito di coppie (v,w) a rappresentare gli archi del grafo, con 0 <= v < V, 0 <= w < V

Si assuma che il grafo non sia un multigrafo e che sia semplice.

Si legga una proposta di partizionamento da file, in un formato a discrezione del candidato, e si valuti se questa sia una **domatic partition** valida, che quindi:
- rappresenti un partizionamento valido dei vertici
- sia tale che ogni sottinsieme sia un dominating set

Si individui la **domatic partition** a cardinalità massima, ossia la partizione composta dal maggior numero possibile di dominating set per il grafo in input.