Algorithms and data structures

Iniziato	domenica, 18 dicembre 2022, 12:31
Stato	Completato
Terminato	domenica, 18 dicembre 2022, 12:32
Tempo impiegato	21 secondi
Valutazione	**0,00** su un massimo di 36,00 (**0**%)

Domanda 1

Risposta non data

Punteggio max.:
1,00

Suppose to have an initially empty priority queue implemented with a maximum heap.

Consider the following sequence of integers and "*" characters, where each integer corresponds to one insertion into the priority queue and each character "*" corresponds to one extraction.

```
1  3  8  7  17  13  *  *
```

Report the sequence of values as they are stored in the array representing the priority queue at the end of the entire process.

Please, show the entire content of the array as a sequence of integer values separated by a single space. No other symbols must be included in the response. This is an example of the response: 0 3 2 6 8 etc.

Risposta: ☒

La risposta corretta è : 8 7 3 1

Domanda 2

Risposta non data

Punteggio max.:
9,00

A password is a palindrome string of 7 characters, where:

- The first and seventh characters are the same small letter (a, ..., z).
- The second and sixth characters are the same capital letter (A, ..., Z).
- The third and fifth characters are the same digit (0, ..., 9).
- The fourth character is always a '-' or a '+' character.

For example, passwords aA1-1Aa, cZ9+9Zc, and fL0+0Lf are acceptable, whereas aA1+1Ab is not.

Write a recursive function to generate all passwords satisfying the previous constraints and write them on standard output.

First, define the main property which differentiates a binary tree from a BST.

Then, suppose a BST storing strings of unknown size is given. Each BST node includes a pointer to the parent and the size of the BST rooted at that node. Write the function to insert a new key into the leaves of the BST. The node type and the function prototypes are the following.

```
typedef struct node_s {
    char *key;
    int size;
    struct node_s *parent;
    struct node_s *left;
    struct node_s *right;
} note_t;

node_t *insert (node_t *root, char *key);
```

Pay particular attention to updating the size of all visited nodes and setting the pointer to the parent of the newly inserted node. If necessary, add parameters to the function insert.

A BST contains integer values included in the range 1-1000. Suppose we are looking for the value 711 is such a BST. Consider the following sequences of values generated during a search.

```
524  565  626  860  681  785  714  693  699  700  715  713  711
662  938  689  884  691  840  771  717  699  714  707  712  711
```

Indicate which sequences (numbered as 1 and 2) are correct.

Please, report the numbers indicating the correct sequences. These numbers must be separated by a single space and reported in ascending order. No other symbols must be included in the response. This is an example of the response format (when both sequences are correct): 1 2.

Risposta: ✗

La risposta corretta è : 2

Domanda 5

Risposta non data

Punteggio max.: 2,00

Analyze the following program. Indicate the exact output generated.

Please, report the exact program output with no other symbols.

```c
#define R 3
#define C 5

int f (int [][C]);

int main(void) {
  int mat[R][C] = {
    {1,2,3,4,5},
    {2,3,4,5,6},
    {3,4,5,6,7}
  };
  printf ("%d", f(mat));
  return (1);
}
int f (int mat[R][C]) {
  int i, j, s, m;
  for (i=0; i<R; i++) {
    s = 0;
    for (j=0; j<C; j++) {
      s = s + mat[i][j];
    }
    if (i==0 || s>m) {
      m = s;
    }
  }
  return (m);
}
```

Risposta: ✗

La risposta corretta è : 25

Analyze the following recursive program. Indicate the exact output generated.

Please, report the exact program output with no other symbols.

```
int f (int n, int vet[], int i) {
  int q;
  if (n<=0) {
    return i;
  }
  q = n/2;
  vet[i++] = n%2;
  return (f (q, vet, i));
}

int main () {
  int vet[10], i, n;
  n = f(13, vet, 0);
  for (i=n-1; i>=0; i--)
    fprintf (stdout, "%d", vet[i]);
  return 1;
}
```
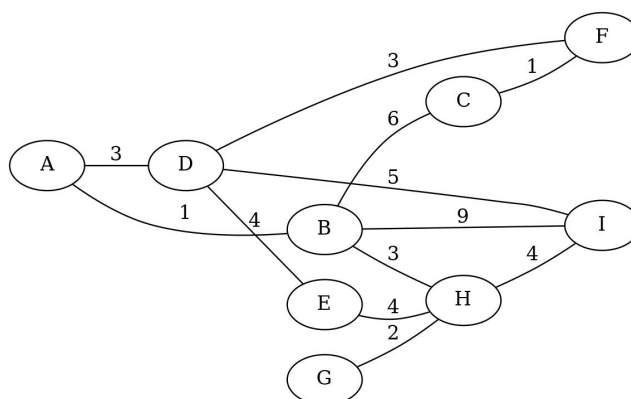
Risposta: ✗

La risposta corretta è : 1101

Given the following undirected and weighted graph find a minimum spanning tree using Prim algorithm. Start from vertex A.

Indicate the total weight of the final minimum spanning tree. Report one single integer value. No other symbols must be included in the response. This is an example of the response format: 13

Risposta: ✗

La risposta corretta è : 21

Domanda 8

Risposta non data

Non valutata

If you want to withdraw from the exam, please select true. Otherwise, i.e., you want to take the exam, select false.

○ (a) False (No, I do not want to withdraw)

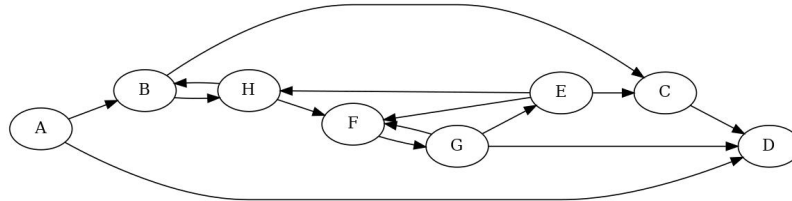○ (b) True (Yes, I want to withdraw)

La risposta corretta è: True (Yes, I want to withdraw)

Domanda 9

Risposta non data

Punteggio max.: 1,00

Given the following directed graph, represent the reverse graph and find all strongly connected components.

Compute all strongly connected components in the graph. Report them in alphabetic order and within each component indicates vertices in alphabetic order (i.e., ACZ and not CAZ or ZAC). Separate components with a single space. No other symbols must be included in the response. This is an example of the response format: ABC EF XY etc.

Risposta: [   ] ✗

La risposta corretta è : A BEFGH C D

Domanda 10

Risposta non data

Punteggio max.: 1,00

In an activity set, the i-th activity is identified by the pair **[si, fi)**, where **si** is the starting time and **fi** is the finishing time. The activities are numbered starting from 1.
The following is a correct set of activities.

```
P1   7   9
P2   17  19
P3   4   6
P4   15  16
P5   6   8
P6   3   7
P7   29  31
P8   28  30
P9   9   12
P10  10  11
```

Using a greedy algorithm, find the largest subset of mutually compatible activities. Please, report the set of compatible activities in the same order they have been selected separated by a single space. No other symbols must be included in the response. This is an example of the response: 1 3 2 6 8

Risposta: [   ] ✗

La risposta corretta è : 3 5 10 4 2 8

Write the function

> void eraseDuplicate (char *str);

that erase all duplicated characters in the string **str**.
More specifically, given the string **str,** the function **eraseDuplicate** modifies the same string such that for all characters appearing in the string more than once, only the first occurrence is retained.

For example, if the string **str** initially contains

> aa;;;bbbab;

when the function returns, it must contain

> a;b

Consider a binary tree whose visits return the following sequences.

Pre-order:	A	B	G	I	C	E	F	L	D	H
In-order:	I	G	B	A	F	E	L	C	D	H
Post-order:	I	G	B	F	L	E	H	D	C	A

Report the sequence of keys stored on the leaves of the tree, moving on the tree from left to right. Please, report the list of keys on the same line, separated by a single space. No other symbols must be included in the response. This is an example of the response format: A X C Y etc.

Risposta: ❌

La risposta corretta è : I F L H

Write the function

Consider the following implementation of heapsort.

```
void heap_sort (heap_t *heap) {
  int i, size;
  build_heap (heap);
  size = heap->size;
  for (i=heap->size-1; i>0; i--) {
      swap (heap, 0, i);
      heap->size--;
      heapify (heap, 0);
  }
  heap->size = size;
  return;
}

static void build_heap (heap_t *heap) {
  int i;
  for (i=(heap->size>>1)-1; i>0; i--) {
      heapify (heap, i);
  }
  return;
}

static void heapify (heap_t *heap, int i) {
  int l, r, m;
  l = LEFT(i);
  r = RIGHT(i);
  if ((l<heap->size) && (heap->v[l]<heap->v[i]))
    m = l;
  else
    m = i;
  if ((r<heap->size) && (heap->v[r]<heap->v[m]))
    m = r;
  if (m != i) {
    swap (heap, i, m);
    heapify (heap, m);
  }
  return;
}
```

Indicate which of the following statements are correct.

Note that more than one response can indeed be correct and that incorrect answers may imply a penalty on the final score.

___

Scegli una o più alternative:

- ☐ (a) In function **heapify** the variable **m** indicates the position of the smallest among three elements of the array.

- ☐ (b)
  The cycle **for** inside **build_heap** has to go until **l>=0** not until **l>0**.

- ☐ (c)
  The variable **size** inside the heap structure must be accessed as **heap.size**, not as **heap->size**, everywhere.

- ☐ (d) The **heapify** procedure has a linear cost.

- ☐ (e) It sorts the array in ascending order.

- ☐ (f)
  The cycle for inside **heap_sort** must start from **i=heap->size**, not **i=heap->size-1**.

La risposta corretta è:
The cycle **for** inside **build_heap** has to go until **l>=0** not until **l>0**., In function **heapify** the variable **m** indicates the position of the smallest among three elements of the array.

Write a C function to insert registry data in the correct position in an ordered list.
Data include the surname and name of a person; both fields are strings of characters of maximum length equal to 20.
Use the surname as the primary ordering key, and the name as the secondary ordering key.

The prototype of the function is the following:

```
int orderInsert (list_t *list, char *surname, char *name);
```

The function receives the list pointer (of type **list_t**) and the two strings storing the surname and the name of the new person. The function returns an integer value that is equal to zero if the person is already present in the list, and equal to one if the insertion operation succeeds.

Define the type **list_t**, considering that, for each person, surname and name have to be copied in new **dynamically allocated** and **separated** fields.

Given the following array of integer values, sort it in ascending order using the merge sort procedure.

```
13  11  8  6  3  2  9  10  1  15
```

Display the content of the array just before the last (and conclusive) merge step (the one delivering the final and sorted array). Please, show the entire content of the array as a sequence of integer values separated by a single space. No other symbols must be included in the response.

Risposta: ✗

La risposta corretta è : 3 6 8 11 13 1 2 9 10 15

Insert the following sequence of keys into an initially empty hash table. The hash table has a size equal to M=23. Insertions occur character by character using open addressing with double hashing. Use function h1(k)=k%M and h2(k)=1+(k%97).
Each character is identified by its index in the English alphabet (i.e., A=1, ..., Z=26). Equal letters are identified by a different subscript (i.e., A and A become A1 and A2).

```
A  F  A  F
```

Indicate in which elements are placed the last two letters of the sequence, i.e., A and F, in this order. Please, report your response as a sequence of integer values separated by one single space. No other symbols must be included in the response. This is an example of the response format: 3 9
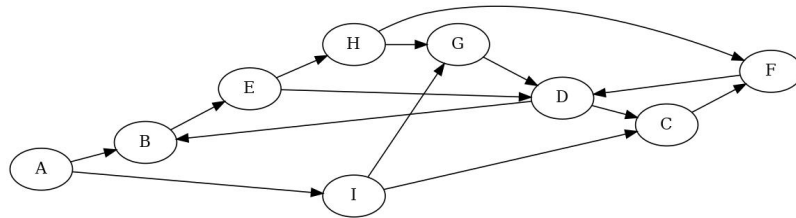
Risposta: ✗

La risposta corretta è : 3 13

Visit the following graph in depth-first, starting at node A.

Label each edge as tree (T), back (B), forward (F), and cross (C). When necessary, consider nodes and edges in alphabetic order.

Report the label of the edges FD, AI, and IC in this order. Please, indicate the edge type of these 3 arcs with a single letter, i.e., T, F, B, C, separated by single spaces. No other symbols must be included in the response. This is an example of the response: F T B

Risposta: ✗

La risposta corretta è : B T C