

Politecnico
di Torino

Algorithms and Programming

Iniziato martedì, 21 dicembre 2021, 17:29**Stato** Completato**Terminato** martedì, 21 dicembre 2021, 17:57**Tempo impiegato** 28 min. 28 secondi**Valutazione** 0,00 su un massimo di 40,00 (0%)**Domanda 1**

Risposta non data

Punteggio max.:
2,00

THIS EXERCISE IS ONLY FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS (NOT FOR COMPUTER ENGINEERING STUDENTS).

Given the following sequence of integers stored in an array, sort it in ascending order using counting sort.

2 3 2 4 10 9 3 0 6 3 4 9 7 10 2

Which is the value of element C[2] at the end of the sorting process?

Risposta: ✖

La risposta corretta è : 1

Domanda 2

Risposta non data

Punteggio max.:
2,00

THIS EXERCISE IS ONLY FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS (NOT FOR COMPUTER ENGINEERING STUDENTS).

Given a binary tree, its visits return the following sequences.

Pre-order S L P H E C D I N M A U K

In-order H E C P I D L N S K U A M

Post-order C E H I D P N L K U A M S

Which are the leaves of the tree? Report them from left to right no matter which their level is within the tree (e.g., A B C etc.).

Risposta: ✖

La risposta corretta è : N I K C

Domanda 3

Risposta non data

Punteggio max.:
2,00

THIS EXERCISE IS ONLY FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS (NOT FOR COMPUTER ENGINEERING STUDENTS).

Insert the following sequence of keys into an initially empty hash table. Each character is identified by its index in the English alphabet ($A=1, \dots, Z=26$). The hash table has a size equal to 23. Insertions occur character by character using open addressing with double hashing. Use function $h_1(h)=k\%M$ and $h_2(h)=1+(k\%97)$.

F A R F R O M H O M E

In which elements are placed the first and the second letter M (e.g., 2 13)?

Risposta:



La risposta corretta è : 4 9

Domanda 4

Risposta non data

Punteggio max.:
2,00

THIS EXERCISE IS ONLY FOR COMPUTER ENGINEERING STUDENTS (NOT FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS).

Using a greedy algorithm find an optimal Huffman code for the following symbols with the specified frequencies.

A:21 B:13 C:14 D:8 E:7 F:16 G:9 H:2 I:17 J:4 K:5 L:12

Report the final code for each letter. Use the following format for your response:
A(0001), B(010001), etc.

Domanda 5

Risposta non data

Punteggio max.:
2,00

Given the following sequence of integers stored into an array, turn it into a heap, assuming to use an array as underlying data structure. Then, execute the first two steps of heapsort on the heap built at the previous step. Assume that, at the end, the largest value is stored at the heap's root.

15 6 19 7 8 6 1 3 22 15 11 17 12 13 33

Report the final content of the entire array as a sequence of values (e.g., 12 3 4 89 etc.).

Risposta:



La risposta corretta è : 19 15 17 7 11 13 15 3 6 8 1 6 12 22 33

Domanda 6

Risposta non data

Punteggio max.:
2,00

Suppose to have an initially empty priority queue implemented with a maximum heap. Given the following sequence of integers and "*" characters, where each integer corresponds to an insertion into the priority queue and each character "*" corresponds to an extraction.

17 2 9 13 7 23 18 3 9 11 5 1 * * *

Report the sequence of values as they are stored in the array representing the priority queue at the end of the entire process (e.g, 3 45 34 etc.).

Risposta:



La risposta corretta è : 13 11 9 9 7 5 1 2 3

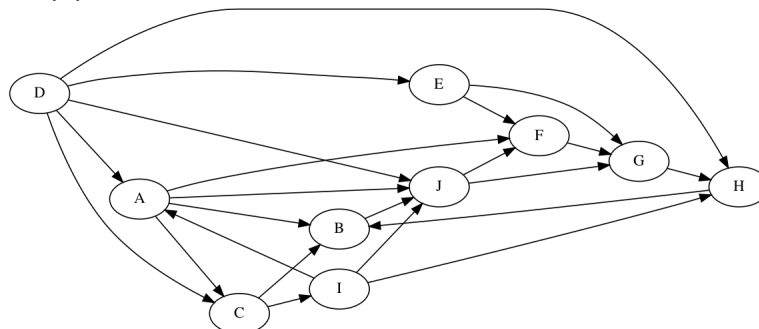
Domanda 7

Risposta non data

Punteggio max.:
1,00

Visit the following graph in depth-first, starting at node A. Label nodes with discovery and end-processing times and edges as T (tree), B (back), F (forward), C (cross). Start with the discovery time set to 1 on A. When necessary, consider nodes and edges in alphabetic order.

Report the labels of all edges by listing for each edge its label. Use the format: AB(T), CD(C), etc.



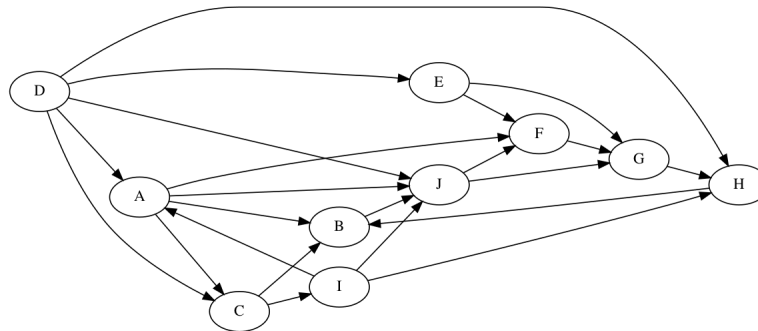
Domanda 8

Risposta non data

Punteggio max.:
1,00

Visit the following graph in depth-first, starting at node A. Label nodes with discovery and end-processing times and edges as T (tree), B (back), F (forward), C (cross). Start with the discovery time set to 1 on A. When necessary, consider nodes and edges in alphabetic order.

Which are the discovery and the end-processing time of E (e.g. 13 27)?



Domanda 9

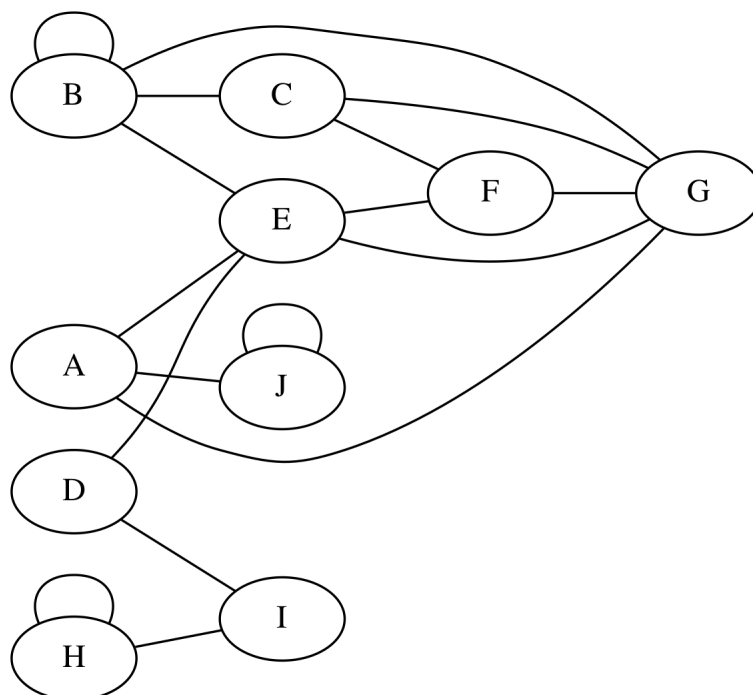
Risposta non data

Punteggio max.:
1,00

THIS EXERCISE IS ONLY FOR COMPUTER ENGINEERING STUDENTS (NOT FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS).

Given the following graph find all articulation points and all bridges. If necessary, consider nodes and edges in alphabetical order.

List alphabetically all articulation points, e.g., B F E etc.



Risposta:



La risposta corretta è : A D E I

Domanda 10

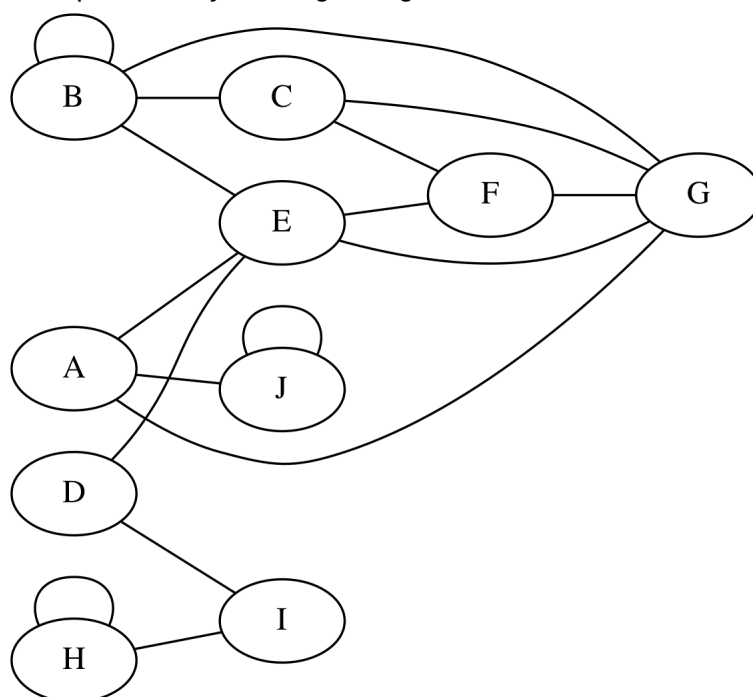
Risposta non data

Punteggio max.:
1,00

THIS EXERCISE IS ONLY FOR COMPUTER ENGINEERING STUDENTS (NOT FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS).

Given the following graph find all articulation points and all bridges. If necessary, consider nodes and edges in alphabetical order.

List alphabetically all bridges, e.g., AC AZ BE etc.



Risposta:



La risposta corretta è : AJ DI ED IH

Domanda 11

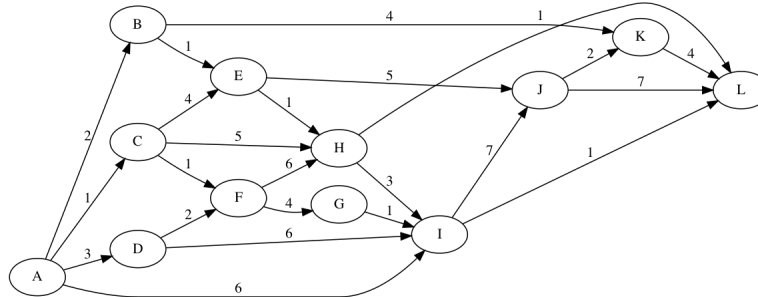
Risposta non data

Punteggio max.:
1,00

THIS EXERCISE IS ONLY FOR COMPUTER ENGINEERING STUDENTS (NOT FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS).

Given the following weighted DAG, find the topological order of its vertices, and then all longest paths from vertex A, using the simplified algorithm for directed acyclic graphs. If necessary, consider nodes in alphabetical order.

Report the topological order of the vertices, e.g., B F E etc.



Risposta:



La risposta corretta è : A D C F G B E H I J K L

Domanda 12

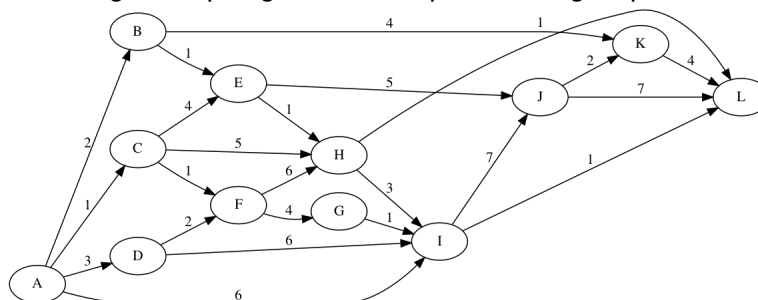
Risposta non data

Punteggio max.:
1,00

THIS EXERCISE IS ONLY FOR COMPUTER ENGINEERING STUDENTS (NOT FOR ELECTRONIC AND COMMUNICATION ENGINEERING STUDENTS).

Given the following weighted DAG, find the topological order of its vertices, and then all longest paths from vertex A, using the simplified algorithm for directed acyclic graphs. If necessary, consider nodes in alphabetical order.

Following this topological order, report the longest path to all vertices, e.g., 9 4 12 3 etc.



Risposta:



La risposta corretta è : 0 3 1 5 9 2 5 11 14 21 23 28

Domanda 13

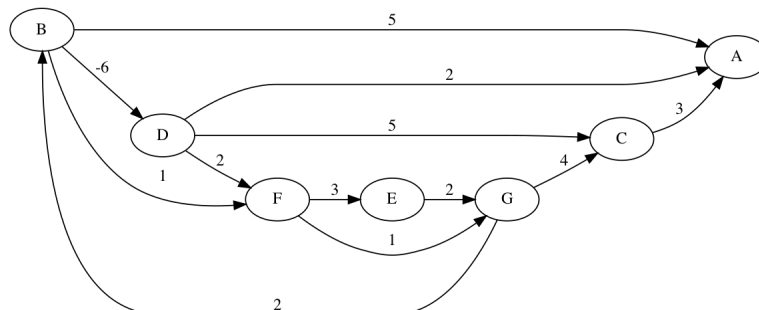
Risposta non data

Punteggio max.:
2,00

Given the following directed and weighted graph, apply Bellman-Ford's algorithm to find all shortest paths connecting node F with all the other nodes.

Consider nodes and edges in alphabetical order.

Report the entire table built to compute the distance of all nodes at each iteration of the algorithm. Indicate whether the algorithm can be applied or not on the graph. In case of a positive answer, report the final shortest path to each node.



Domanda 14

Risposta non data

Punteggio max.:
2,50

The following function receives as parameters, s1 equal to "ABCDEFABCDEABCDABCABA" and s2 equal to "ABC".

```
void f (char *s1, char *s2) {
    int i, j, flag;

    i = 0;
    while (i <= strlen (s1) - strlen (s2)) {
        j = 0;
        flag = 1;
        while (j<strlen (s2) && flag==1) {
            if (*(s1+i+j) != *(s2+j)) {
                flag = 0;
            }
            j++;
        }
        if (flag == 1) {
            fprintf (stdout, "%d\n", i);
            i = i + strlen (s2);
        } else {
            i++;
        }
    }

    return;
}
```

Which are the values displayed by the function? Report the response as a sequence of values, e.g., 4 5 6 etc.

Risposta:



La risposta corretta è : 0 6 11 15

Domanda 15

Risposta non data

Punteggio max.:
2,50

Analyze the following function.

```
int f (int n) {  
    int i, res;  
  
    if (n<=0)  
        return (1);  
  
    res = 0;  
    for (i=0; i<2; i++)  
        res += f(n-i-2) * f(n-i-3);  
  
    return res;  
}
```

Which is the value returned by the function when it is called with n = 4?

Risposta: 

La risposta corretta è : 6

Domanda 16

Risposta non data

Punteggio max.:
3,00

A graph is represented using an adjacency list and visited in depth-first. The following piece of code reports the data structures required to define the graph, and the function used to visit it in depth-first.

```
struct graph_s {
    vertex_t *g;
    int nv;
};

struct edge_s {
    int weight;
    vertex_t *dst;
    edge_t *next;
};

struct vertex_s {
    int id;
    int color;
    int dist;
    int disc_time;
    int endp_time;
    int scc;
    vertex_t *pred;
    edge_t *head;
    vertex_t *next;
};

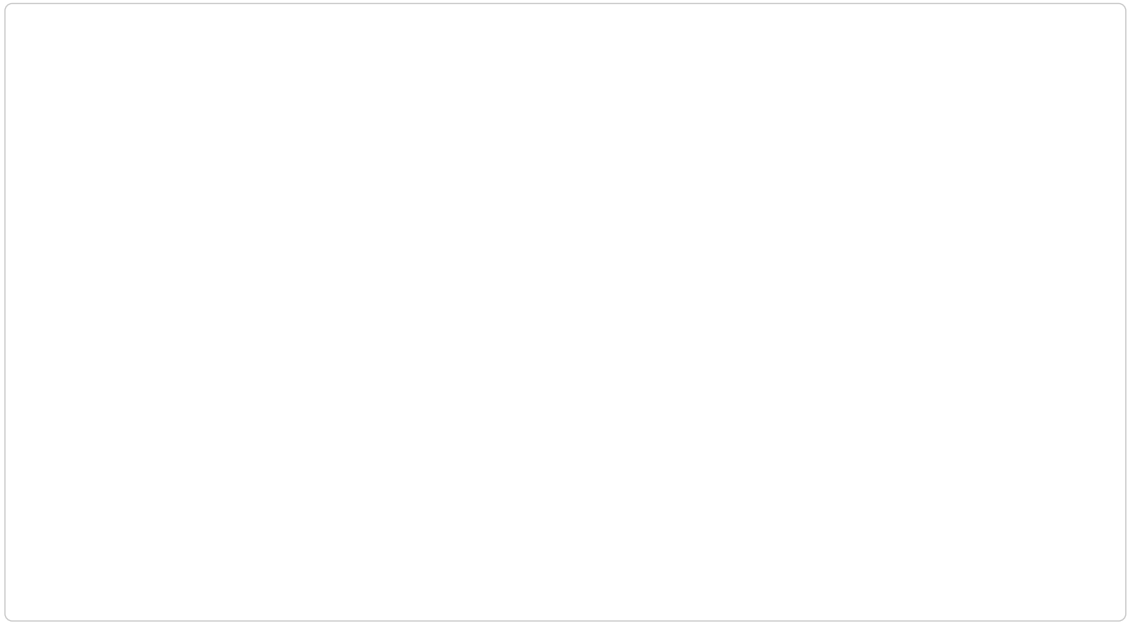
void graph_dfs_r (graph_t *g, vertex_t *n, int currTime) {
    edge_t *e;
    vertex_t *tmp;

    n->color = GREY;
    n->disc_time = ++currTime;

    e = n->head;
    while (e != NULL) {
        tmp = e->dst;
        if (tmp->color == WHITE) {
            tmp->pred = n;
            currTime = graph_dfs_r (g, tmp, currTime);
        }
        e = e->next;
    }
    n->color = BLACK;
    n->endp_time = ++currTime;

    return;
}
```

Modify the function `graph_dfs_r` to complete the following tasks: 1) Compute the number of tree, forward, backward and cross edges in the graph, and 2) compute the depth of the depth-first visit tree.



Domanda 17

Risposta non data

Punteggio max.:
5,00

A heap is represented as a binary tree in which all levels but the last one are complete. The following piece of code defines the data structure on which the heap is based:

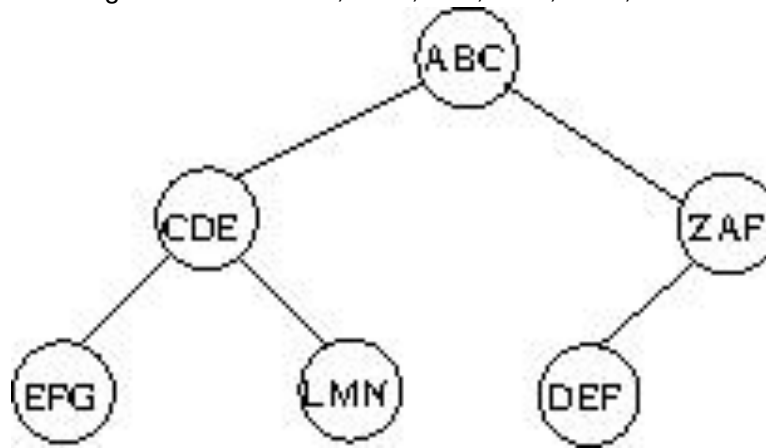
```
typedef struct node_s node_t;  
  
struct node_s {  
    char *val;  
    node_t *left;  
    node_t *right;  
};
```

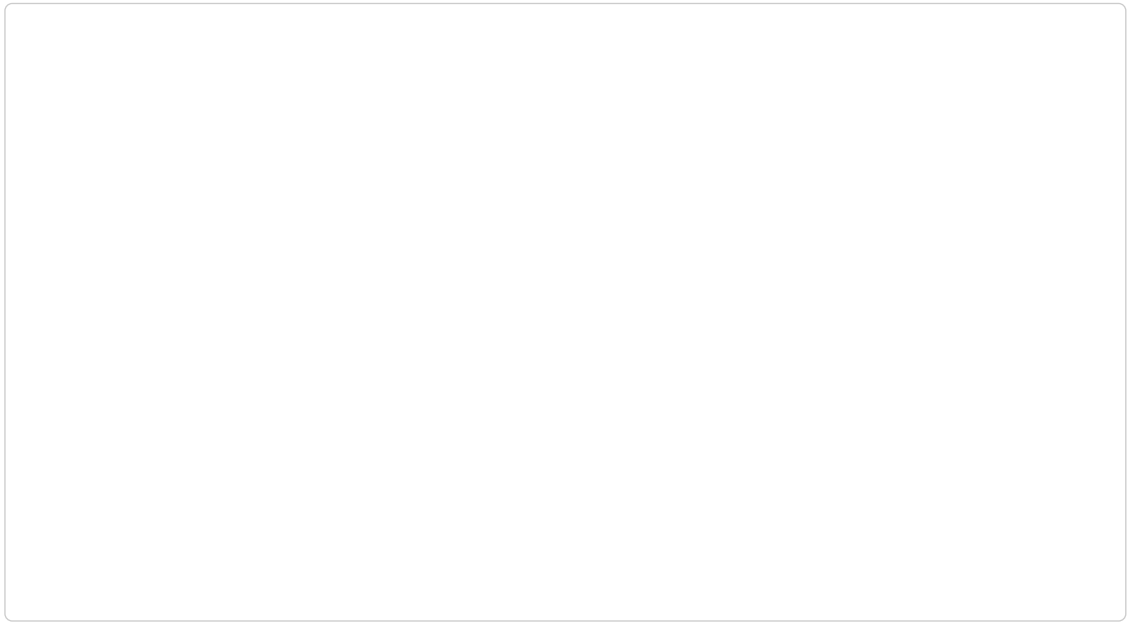
and the following function visits the heap, printing the key of all its nodes:

```
void heap_print (node_t *root) {  
    if (root == NULL)  
        return;  
  
    heap_print (root->left);  
    fprintf (stdout,"%s\n", root->val);  
    heap_print (root->right);  
  
    return;  
}
```

For example for the following heap the function displays the value in the following order: EFG, CDE, LMN, ABC, DEF, ZAF.

Modify the function to display the heap on a level-by-level basis, that is, using the following node order: ABC, CDE, ZAF, EFG, LMN, DEF.





Domanda 18

Risposta non data

Punteggio max.:

7,00

A rectangular matrix of R rows and C columns, contains only capital letters and the '*' character. We want to partition the matrix in subsets of elements such that all subsets contains the same set of capital letters (number and type). Moreover, we want to consider the character '*' as equivalent to any capital letter. For example the set of three letters {'F', 'O', 'O'} is equivalent to the set {'O', 'F', 'O'}, and it is also equivalent to the set {'*', 'O', 'F'}, as the character '*' can "substitute" the letter 'O'. Moreover, given the matrix of Figure (a), with $R=3$ and $C=4$, Figure (b), (c), and (d), show three correct partitions of the matrix, as each one of the 3 subsets (highlighted with different backgrounds) include the 4 characters {'F', 'O', 'O', '*'} (where the '*' is equivalent to any letter). At the same time, also Figure (d) is a correct partition of the matrix, as it includes 4 subsets, each one including the letters {'F', 'O', 'O'} (and in the set {'*', '*', '*'} any star can become any letter). Write the function `find_partition` finding all correct partitions on the matrix `m`, of size $R=r$ and $C=c$.

The function must display all correct partitions of the matrix, i.e., all partitions where all subsets include the same set of letters. For each solution, the function must display the row and column indices of each element within the set. For example, for Figure (b) the function must display the coordinates of the 12 elements of the matrix divided into the 3 subsets highlighted in the figure, i.e., $\{(0,0),(0,1),(0,2),(0,3)\}, \{(1,0),(1,1),(1,2),(1,3)\}, \{(2,0),(2,1),(2,2),(2,3)\}$.

| | | | |
|---|---|---|---|
| F | O | * | O |
| * | F | O | O |
| O | * | O | F |

(a)

| | | | |
|---|---|---|---|
| F | O | * | O |
| | | | |
| | | | |

(b)

| | | | |
|---|---|---|---|
| F | O | * | O |
| | | | |
| | | | |

(c)

| | | | |
|---|--|--|--|
| F | | | |
| | | | |
| | | | |

(d)

...

| | | | |
|---|---|--|--|
| F | O | | |
| | | | |
| | | | |

(e)