

Резултати от малко контролно II на гр. II

| | | Група | Зад 1 | Зад 2 | Зад 3 | Зад 4 | Общо |
|--------|-------------------------------|-------|-------|-------|-------|-------|------|
| 81060 | Деян Георгиев Деянов | 2-ра | | | | | |
| 81090 | Тодор Димов Димов | 2-ра | 0 | 2 | 1 | 1 | 4 |
| 81106 | Божимир Валентинов Маринов | 2-ра | | | | | |
| 81114 | Теодора Антонова Ангелова | 2-ра | | | | | |
| 81122 | Теодор Емилов Тошков | 2-ра | 6 | 2 | 7 | 1 | 16 |
| 81130 | Кристиян Петров Кьосев | 2-ра | 6 | 2 | 0 | 1 | 9 |
| 81138 | Яна Валентинова Мишкова | 2-ра | 1 | 2 | 2 | 0 | 5 |
| 81147 | Михаела Динкова Караиванова | 2-ра | 6 | 2 | 1 | 0 | 9 |
| 81155 | Мартин Руменов Терзийски | 2-ра | 4 | 4 | 0 | 4 | 12 |
| 81173 | Жан Христов Петков | 2-ра | | | | | |
| 81181 | Яница Димитрова Велевска | 2-ра | | | | | |
| 81183 | Мелания Стоянова Бербатова | 2-ра | 6 | 4 | 1 | 3 | 14 |
| 81197 | Александра Валентинова Матева | 2-ра | | | | | |
| 81202 | Моника Ефтимова Ефтимова | 2-ра | | | | | |
| 81205 | Николай Красимиров Николов | 2-ра | | | | | |
| 81221 | Таня Василева Сосова | 2-ра | | | | | |
| 81236 | Ясен Ясенов Висулчев | 2-ра | | | | | |
| 855250 | Магдалена Тришанска | 2-ра | 0 | 4 | 0 | 1 | 5 |
| 81204 | Стоян Иванов Тодоров | 1-ва | 6 | 2 | 1 | 0 | 9 |

Тези, които нямата точки или са правили контролното с друга група, или не са правили изобщо.

Забележки:

Относно задачата с брой пътища, алгоритъм сходен на *DFS*, който търси всички пътища има сложност $O(S)$, където S е броят на пътищата. Те могат да са $O(|V|^2)$, а същевременно ребрата да са $O(|V|)$, което не отговаря на условието.

За да решите така задачата трябва в допълнителен масив да запазвате стойности на вече посетени върхове, вместо отново да търсите рекурсивно (в термините на динамичното програмиране, това се нарича мемоизация).

Другото наблюдение, е че тъй като в графа няма цикли, няма смисъл да проверяваме за вече посетени върхове по пътя (масив `visited`).

Нещо такова:

```
int D[n] = new Array of (-1);  
D[t] = 1;  
  
DFS_visit (node u)  
{  
    if (D[u] != -1) return D[u];  
  
    D[u] = 0;  
  
    for (node v in Adj(u)) D[u] += DFS_visit(v);  
  
    return D[u];  
}
```

Съвет: Ако имате задача за *DAG*, първо преценете дали топологично сортиране няма да помогне за решението.