Пошук в глибину

(DFS)

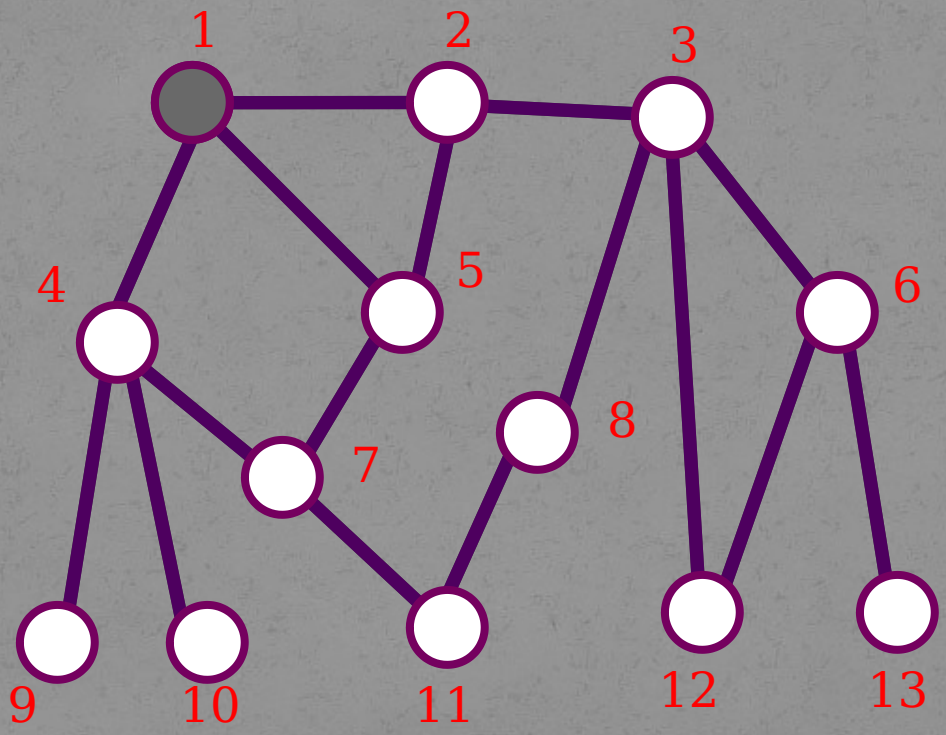
Пошук в глибину

Пошук в глибину (DFS, depth-first search) - являє собою класичний гнучкий алгоритм, який застосовується для розв'язання задачі пов'язаності і безлічі інших завдань обробки графів

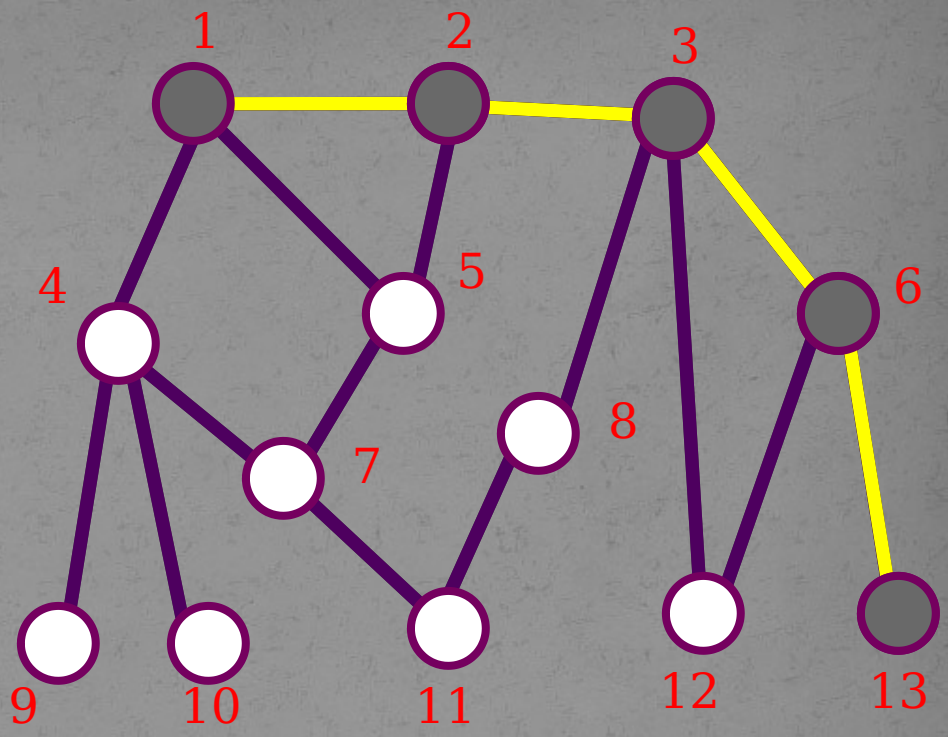
Стратегія пошуку в глибину така: йти "вглиб", поки це можливо (тобто існують не пройдені вхідні ребра), і повертатися і шукати інший шлях, коли таких ребер немає. Так робиться, поки не виявлені всі вершини, досяжні з вихідної.

Алгоритм пошуку в глибину використовує кольори вершин. Кожна з вершин спочатку біла. Будучи виявленою, вона стає сірою; вона стане чорною, коли буде повністю оброблена, тобто коли всі суміжні з нею вершини пофарбовані.

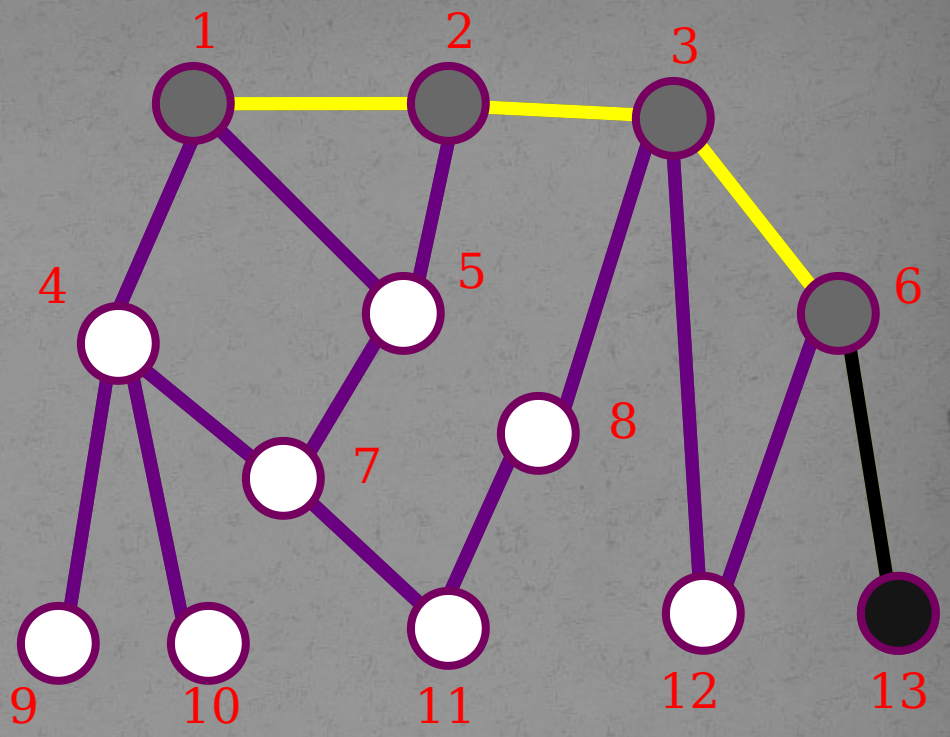
1. Починаємо пошук з довільної вершини s. В якості поточної вершини v беремо вершину s. Фарбуємо її в сірий колір.



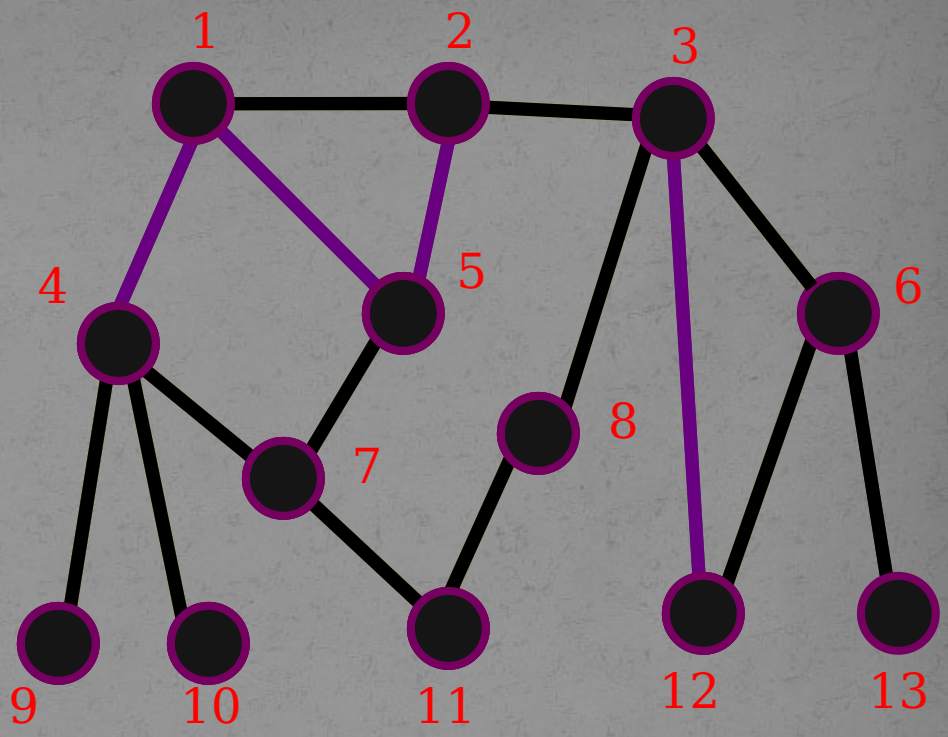
2. З поточної вершини v рухаємося в будь-яку, раніше не пройдену вершину w (білу), якщо така вершина знайдеться (якщо вершини w немає, див. Пункт 3). Запам'ятовуємо дугу, по якій ми потрапили в вершину w. В якості поточної вершини v беремо вершину w.



3. Якщо з вершини v ми не можемо потрапити в раніше не пройдену вершину w, то повертаємося в вершину x, з якої ми потрапили в v, і фарбуємо v в чорний колір. В якості поточної вершини v беремо вершину x.



4. Процес пошуку (пункти 2, 3) закінчується, коли ми намагаємося повернутися назад з вершини, з якої почався пошук (вершина s).



Обхід графа завершено.

Реалізація пошуку (dfs function)

```
int dfs(int** graph, const int n, int from, const int to, int* visited, int* path) {
    if (visited[from] == true)
        return -1;
    visited[from] = true;
    if (from == to) {
        path[0] = to;
        return 0;
    }
    for (int i = 0; i < n; ++i) {
        if (graph[from][i] == 0)
            continue;
        int path_size = dfs(graph, n, i, to, visited, path);
        if (path_size < 0)
            continue;
        path[path_size + 1] = from;
        return path_size + 1;
    }
    return -1;
}
```