

Алгоритм A^*

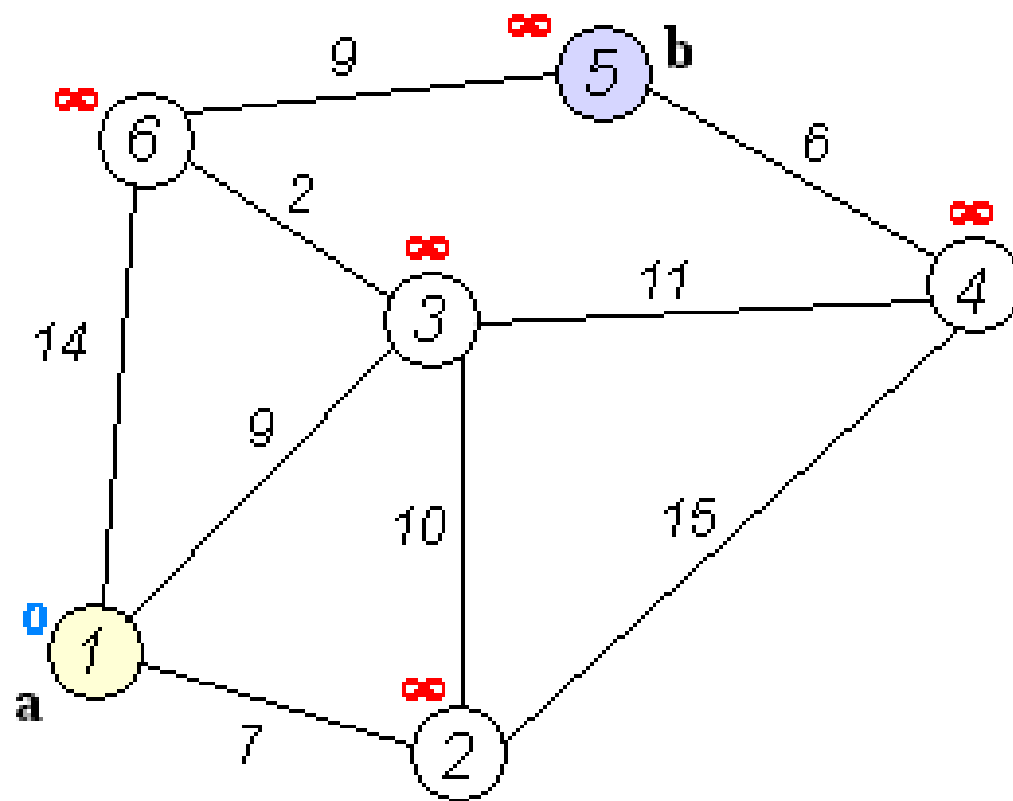
Иван Похабов

30.11.2023

Зачем

- Игры
- Навигаторы, онлайн карты и тому подобное
- Что-то более абстрактное

Алгоритм Дейкстры



Псевдокод

```
vector<int> d(n, inf);  
d[s] = 0;  
q.insert({0, s});  
while (!q.empty()) {  
    int v = q.begin()->second;  
    q.erase(q.begin());  
    for (auto [u, w] : g[v]) {  
        if (d[u] > d[v] + w) {  
            q.erase({d[u], u});  
            d[u] = d[v] + w;  
            q.insert({d[u], u});  
        }  
    }  
}
```

Жадный поиск

- Граница расширяется к цели больше, чем к другим направлениям.
- Заменяем пройденное расстояние на результат эвристической функции.
- Визуализация:
 - <https://www.youtube.com/watch?v=pWaFhNeaUJw&t=15s>
 - <https://www.youtube.com/watch?v=8ALNM6bYyGY&t=4s>

Эвристическая функция

- Должна быть:

1. Монотонной(преемственной)
2. Допустимой

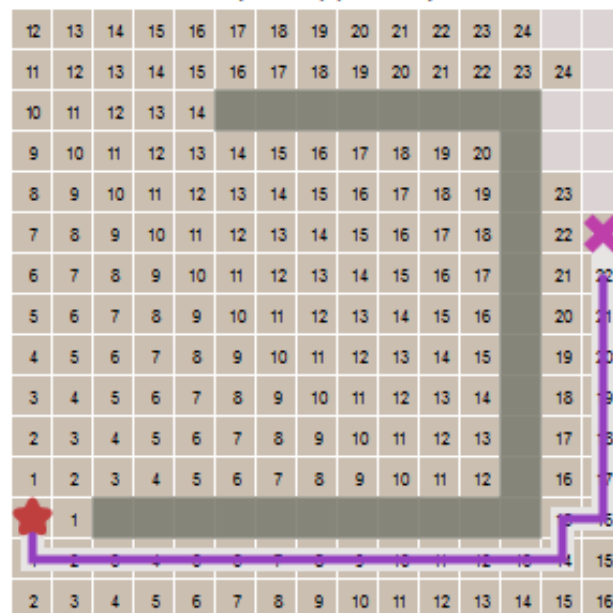
- Примеры:

1. Манхэттенское расстояние
2. Расстояние Чебышева
3. Евклидово расстояние по прямой

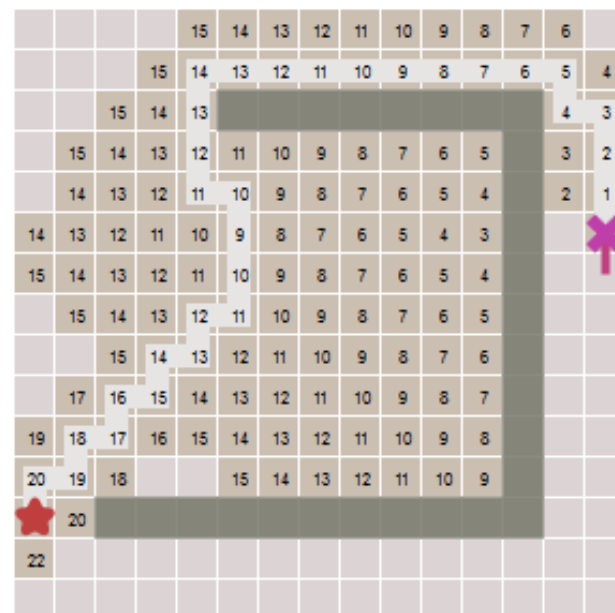
Алгоритм A^*

- Соединим алгоритм Дейкстры и алгоритм Жадного поиска. Теперь в алгоритме мы будем брать вершину с наименьшей суммой уже пройденного расстояния и предполагаемого оставшегося расстояния (значение допустимой монотонной эвристической функции).
- Визуализация и сравнение с Дейкстрой:
- <https://www.youtube.com/watch?v=9REexHx0hDY>

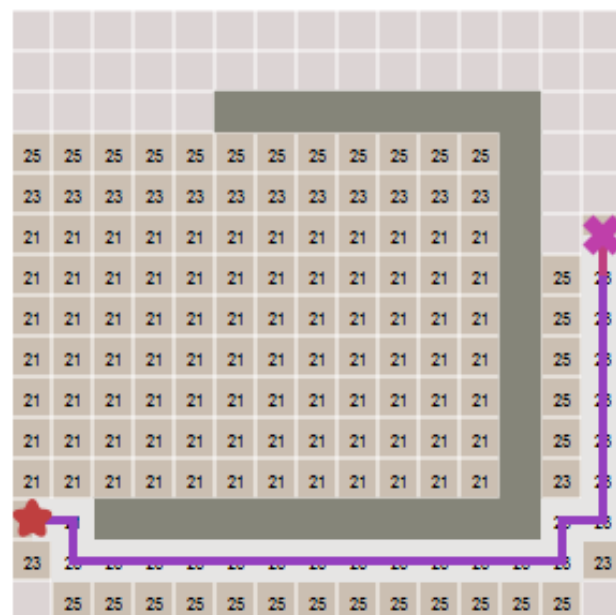
Алгоритм Дейкстры



Жадный поиск



Поиск A*



Псевдокод

```
vector<int> d(n, inf);
d[s] = 0;
q.insert({0, s});
while (!q.empty()) {
    int v = q.begin()->second;
    q.erase(q.begin());
    for (auto [u, w] : g[v]) {
        if (d[u] > d[v] + w) {
            q.erase({d[u] + heuristic(u), u});
            d[u] = d[v] + w;
            q.insert({d[u] + heuristic(u), u});
        }
    }
}
```

Полезные ссылки

- <https://www.redblobgames.com/pathfinding/a-star/introduction.html>
- <https://habr.com/ru/articles/331192/> (перевод прошлой статьи)
- https://neerc.ifmo.ru/wiki/index.php?title=%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%A*
- <https://ru.ruwiki.ru/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%94%D0%B5%D0%B9%D0%BA%D1%81%D1%82%D1%80%D1%8B>
- <https://ru.algorithmica.org/cs/shortest-paths/>