1 Задача 8.5

Заведем массив расстояний `dist`, в `dist[v, k]` хранится расстояние от вершины `v` до 2^k-го предка. Предподсчитаем его вместе с двоичными подъемами тамким образом:
`dist[v, k]= dist[v, k-1] + dist[jmp[v, k-1], k-1]` и `dist[v, 0]` расстояние от вершины `v` до родителя.
Будем искать ближайшего предка v растояние до которого не меньше l так:
Если `k = 0`, то мы находимся в вершине, чей предок и есть искомая вершина.
Иначе если `dist[v, k] >= l`, то запускаем поиск от `v, k - 1, l`
Иначе запускаем поиск от `jmp[v, k], k - 1, l - dist[v, k]`

```
search(v, k, l):
  if(k == 0):
    return jmp[v, k];
  if(dist[v, k] >= l):
    return search(v, k-1, l);
  return search(jmp[v, k], k - 1, l - dist[v, k]);
```

Т.к. с каждым шагом `k` уменьшается, то время работы $O(k_0)$, где k_0 - начальное, такое максимальное, что 2^{k_0} не больше количества вершин на пути до корня. Тогда видно, что время работы $O(\log n)$