Algo programming 2

Author

- 108062213 顏浩昀

1. Pseudo code

```
make a struct Node{
    level, vertex, lower_bound, matrix, path
}

Solve(){
    priority_queue pq;
    pq.push(root)

    while(!pq.empty()){
        cur = pq.top();
        index = cur->vertex
        pq.pop();
        for j = (0,N) {
            find a vertex not in cur->path
            create new Node u
            u->level = cur->level+1
            u->path = cur->path
            u->matrix = cur->matrix
            u->lower_bound = cur->lower_bound + cost from cur to u
                                + cal u lowerbound
            pq.push(u);
        }
        delete cur;
    }
}

Cal(){ //calculate lower bound
    visited index row and col set as INT_MAX
    col[j], row[i] store min in col[j] and row[i]
    reduce matrix

    lower_bound += col[j]
    lower_bound += row[i]
}
```

2. Time Complexity

    1. In Cal(), reduce matrix need O(n^2)

2. In Solve(), BFS need O(V+E) = O(n)
3. In main(), standard input need O(n^2)

Totally T(n) = O(n^2)