算法基础
第八次作业（DDL：2024 年 12 月 5 日 23:59）
解答过程中请写出必要的计算和证明过程

Q1. (20 分) 假定图中的边权重全部为整数，且在范围 $1 \sim |V|$ 内。在此种情况下，Kruskal 算法最快能多快？如果边权重取值范围在 1 到某个常数 W 之间呢？

Solution:

- 假定图中的边权重全部为整数，且在范围 $1 \sim |V|$ 内。对边权重排序使用计数排序 (counting sort)，时间复杂度为 $O(E + |V|)$，由于连通图 $|E| \geq |V| - 1$，$O(E + |V|) = O(E)$；并查集操作时间复杂度为 $O(E\alpha(V))$。总时间复杂度为 $O(E + E\alpha(V)) = O(E\alpha(V))$。
- 如果边权重取值范围在 1 到某个常数 W 之间。对边权重排序使用计数排序 (counting sort)，时间复杂度为 $O(E + W)$，由于 W 是常数，$O(E + W) = O(E)$；并查集操作时间复杂度为 $O(E\alpha(V))$。总时间复杂度为 $O(E + E\alpha(V)) = O(E\alpha(V))$。

Q2. (20 分) 修改 Bellman-Ford 算法，使其对于所有结点 v 来说，如果从源节点 s 到结点 v 的一条路径上存在权重为负值的环路，则将 $v.d$ 的值设置为 $-\infty$。

Solution: Bellman-Ford 在标准实现中执行 $|V|-1$ 次松弛操作，能够找到从源结点 s 到所有顶点的最短路径。如果图中存在从 s 可达的负权环，则在执行 $|V|$ 次松弛操作时，这些环上的结点的 $v.d$ 值仍会减少。而若不存在负权环，执行 line1 - line4 后 $v.d$ 不会再减小。所以只需将 line7 替换为 $v.d = -\infty$。

Q3. (30 分) 给定带权重的有向图 $G = (V, E)$，其权重函数为 $w : E \to \{0, 1, 2, \cdots, W\}$，这里 W 为某个非负整数。请修改 Dijkstra 算法来计算从给定源结点 s 到所有结点之间的最短路径。该算法时间应为 $O(WV + E)$。

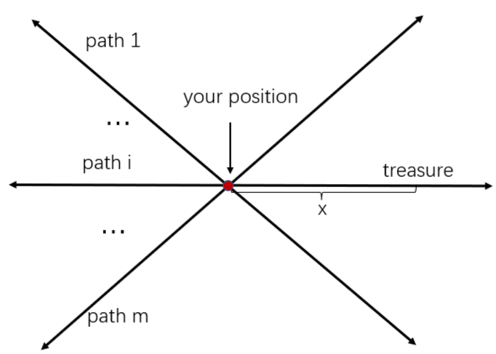
Solution: 利用边权范围有限，优化 Dijkstra 算法中优先队列的操作，将优先队列替换为数组 A，其中 $A[i]$ 保存的是从源节点到其估计距离为 i 的顶点。取值范围是 $[0, WV] \cup \{\infty\}$，因为最坏情况下，我们需要计算从链的一端到另一端的距离，其中链包含 V 个顶点，每个顶点之间通过权重为 W 的边连接，所以 A 的长度设置为 $WV + 2$。此外，我们还需要为每个顶点 u 添加一个属性 $u.list$，指向顶点 u 在链表 $A[u.d]$ 中的位置。由于最小距

Algorithm 1 MODIFIED-DIJKSTRA(G, w, s)

```
for each v ∈ G.V do
    v.d ← VW + 1
    v.π ← NIL
end for
s.d ← 0
Initialize an array A of length VW + 2
A[0].insert(s)
Set A[VW + 1] equal to a linked list containing every vertex except s
k ← 0
for i ← 1 to |V| do
    while A[k] = NIL do
        k ← k + 1
    end while
    u ← A[k].head
    A[k].delete(u)
    for each vertex v ∈ G.Adj[u] do
        if v.d > u.d + w(u, v) then
            A[v.d].delete(v.list)
            v.d ← u.d + w(u, v)
            v.π ← u
            A[v.d].insert(v)
            v.list ← A[v.d].head
        end if
    end for
end for
```

离总是递增的，变量 k 最多只会达到 WV。因此，在第 10 行的 for 循环的所有迭代中，第 11 行的 while 循环总共花费 $O(WV)$ 的时间。在第 16 行的 for 循环中，我们只需遍历每个顶点的邻接表一次，总共花费 $O(V+E)$ 的时间。其他所有操作的时间复杂度为 $O(1)$。因此，算法的总运行时间为 $O(WV)+O(V+E)=O(WV+E)$。

Q4. (30 分) 你站在一个 m 条路的中间汇聚点 (如图所示)，此处有一标识，告知前方某处有一个海盗宝藏目前无人认领。然而，该标志没有说明宝物在哪条路上或哪个方向上，也没有告诉你这里与宝藏之间的距离。你的目标是找到宝藏，同时尽量减少你所耗费的路程。所以如果你走在正确的道路且正确的方向上永远不转身，你的成本是最优解 x (因为你走了 x 距离后会找到宝藏)。但是，如果你走错了方向，从不转身，那么你的成本将是无限的 (因为你会一直走下去)。在这一问题中设 x 为整数。设计一个确定性的竞争比为 $O(m)$ 的算法。并证明这一算法的竞争比。



Solution: 将 2m 个路口依次编号为 $1,2,...,2m$，第 k 次寻找时，依次在每条路上前进 2^{k-1} 的距离，在前进过程中找到宝藏则算法终止，若未找到宝藏。则原路返回，并从路口进入下一条道路，直到 m 条道路的两个方向全部寻找结束。第 k 次寻找若未找到宝藏，则开始进行第 $k+1$ 次寻找。所需要的路程最多是 $8m$。在线算法最坏情况成本在 $x=2^i$ 时取到，

$$2\times 2m(2^0+2^1+\cdots+2^i)=4m(2^{i+1}-1)\leq 8m\times 2^i=8mx \qquad (1)$$