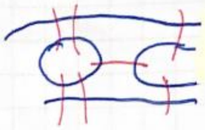
6-1

圖形概論

七橋Q



G={V,E}

V(G): vertex set

E(G): edge set

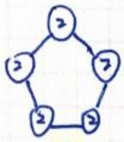
⇒找出最快路法

6-2

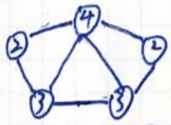
七橋解法

1. degree: 每個點連到邊的個數, 2. vertex types (1) odd or even degrees.

3. Eulerian path (trial) / Euler walk. ① visit every edge exactly once ② 0 or 2 nodes with odd degrees



0 node with odd degrees



2 node

③ what if one node with odd degrees? → 1,3,5,7 奇數個 degree 不会發生

6-3

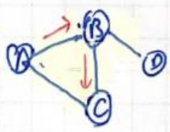
應用於一筆畫

1. Eulerian circuit / cycle / tour (無向圖), 2. Directed Graph (有向圖), 3. Adjacent vertices (相鄰的)

4. Edge is incident to vertices, 5. path: a sequence of edges (路程).

6-11 DFS

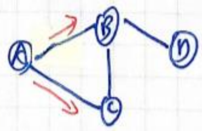
⇒ recursive form ⇒ Has an iterative form that uses a stack



6-12 BFS

(1) iterative form use a queue

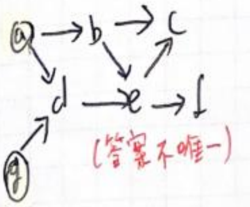
(2) recursive form is possible but not simple



7-1

拓樸排序

1. directed graph without cycle, 2、Acyclic Digraph or Directed Acyclic Graph (DAG)



(答案不唯一)

7-3 拓樸排序的演算版本

① Find a vertex that has no successor (out-degree = 0)

② Add the vertex to the beginning of a list

③ Remove that vertex from the graph, as well as all edges that lead to it.

④ Repeat until the graph is empty

7-5 生成樹

1. A tree is an undirected connected graph without cycle (acyclic)

2. connected, acyclic

⇒ connected ⟶ spanning tree ⟵ acyclic
(邊愈多愈好)　　(臨界邊)　　(邊愈少愈好)

7-6 生成樹特性

1. Detecting a cycle in an undirected connected graph → DFS, BFS

2. A connected undirected graph that has n vertices must have at least n-1 edges (4點3邊)

3. Number of spanning trees

ex How many different spanning trees? ⇒ → isomorphic

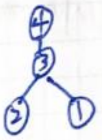
7-7 Prüfer sequence

1、n 點存下來的字串只需 n-2,

2、conversion algorithms.

(1) leaf with the smallest label。

(2) keep the label of its parent

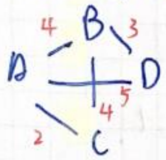


33 (唯一)

7-10 最小生成樹

1、cost of spanning tree

→ sum of the edge weights on a spanning tree.



DFS: 4+4+3=11

BFS: 4+2+5=11

MST: 4+2+3=9

2、a particular graph could have several minimum spanning trees.

3、other variations

① minimum steiner tree (指定 n 個點

② k-minimum spanning tree.

Find a minimum spanning tree that begins at any given vertex.

(1) Find the least-cost edge (v,u) from a visited vertex v to some unvisited vertex u.

(2) Mark u as visited

(3) Add the vertex u and the edge (v,u) to the minimum spanning tree.

(4) Repeat

7-14 初探 shortest 路徑

① shortest path between 2 vertex in a weight graph is the path that has the smallest sum of its edge weights.

② Dijkstra's Algorithm

⇒ Find the shortest paths between a given origin and all other vertices.

⇒ Dijkstra 演算法的應用 → 地圖

L8 圖形問題

8-1 初探 key 路徑分析

1. Activity-on-vertex (AOV) Network → 活動在點上

2. Activity-on-Edge (AOE) Network → 活動在邊上

(1) Directed edge: activity (task) to be performed

(2) vertex: event to signal the completion of certain activit

(3) 不能有 cycle

(4) path length: the total time from start to end.

(5) Critical path: a path with the longest length.

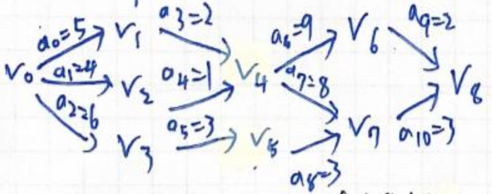
→ minimum time required to complete the project.

8-2

lastest time of an activity: la [0.....10]

→ lastest time of event: le [0.....8]

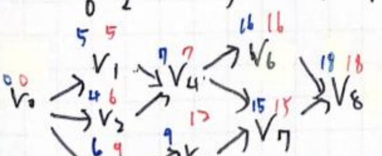
a_1, a_2 可 delay



(1) $la[x] = le[j]$ - duration of $\langle v_i, v_j \rangle$, where a_x is on $\langle v_i, v_j \rangle$

(2) $le[x] = \min\{ le[j]$ - duration of $\langle v_i, v_j \rangle \}$ for every v_j that is an immediate succesor of v_i

la: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]
0 2 3 5 6 9 7 7 12 16 15



[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
0	0	0	5	4	6	7	7	9	16	15
0	2	3	5	6	9	7	7	12	16	15
0	2	3	0	2	3	0	0	3	0	0

(3) la-ea is called (total) float or slack → amount a delay to project completion time

le-ea==0 means a critical activity

(4) Determine critical path.

8-3

1. like topsort (top)

(1) Find vertex v that has no sucessor (out-degree =0)

(2) Add v to the beginning of a list

(3) Remove.

2. ① Find the vertex v that has no predecessor (in-degree = 0)

② For each immediate successor u, do the follow:

→ set ea[x] = ee[v], where x is the activity on <v,u>

→ set ee[u] = max{ ee[u], ee[v] + duration of <v,u> }

→ Decrease the in-degree of u

3. in-degree activity duration

v_0 0 → a_0, 5 → a_1, 4 → a_2 6

v_1 1 → a_3 2

v_2 1 → a_4 1

v_3 1 → a_5 3

v_4 2 → a_6 9 → a_7 6

v_5 1 → a_8 3

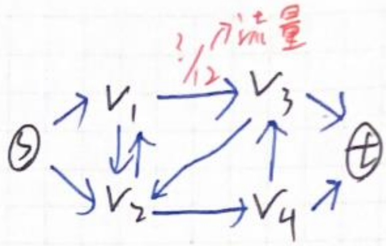
v_6 1 → a_9 2

v_7 2 → a_{10} 3

v_8 2 → Null

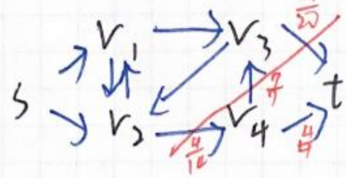
8-5 最大流量 question

1. We are given a flow network G with source s and sink t, and we wish find a flow of maximum value from s to t



flow $f(u,v)$ / capacity $c(u,v)$

(1) single source sink maximum flow problem, (2) maximum-flow, min-cut theorem.



8-7 剩餘圖

1. residual capacity: $c(u,v) = c(u,v) - f(u,v)$

2. Edmonds-Karp algorithm (放大的)

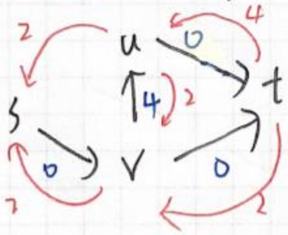
→ Heuristic to find augmenting path

$s \to u \to t$, $c(s,u)=2$, $c(u,t)=4$

→ flow $(u,v)=2$

$s \to v \to t$ flow $(u,v)=2$

⇒



⇒