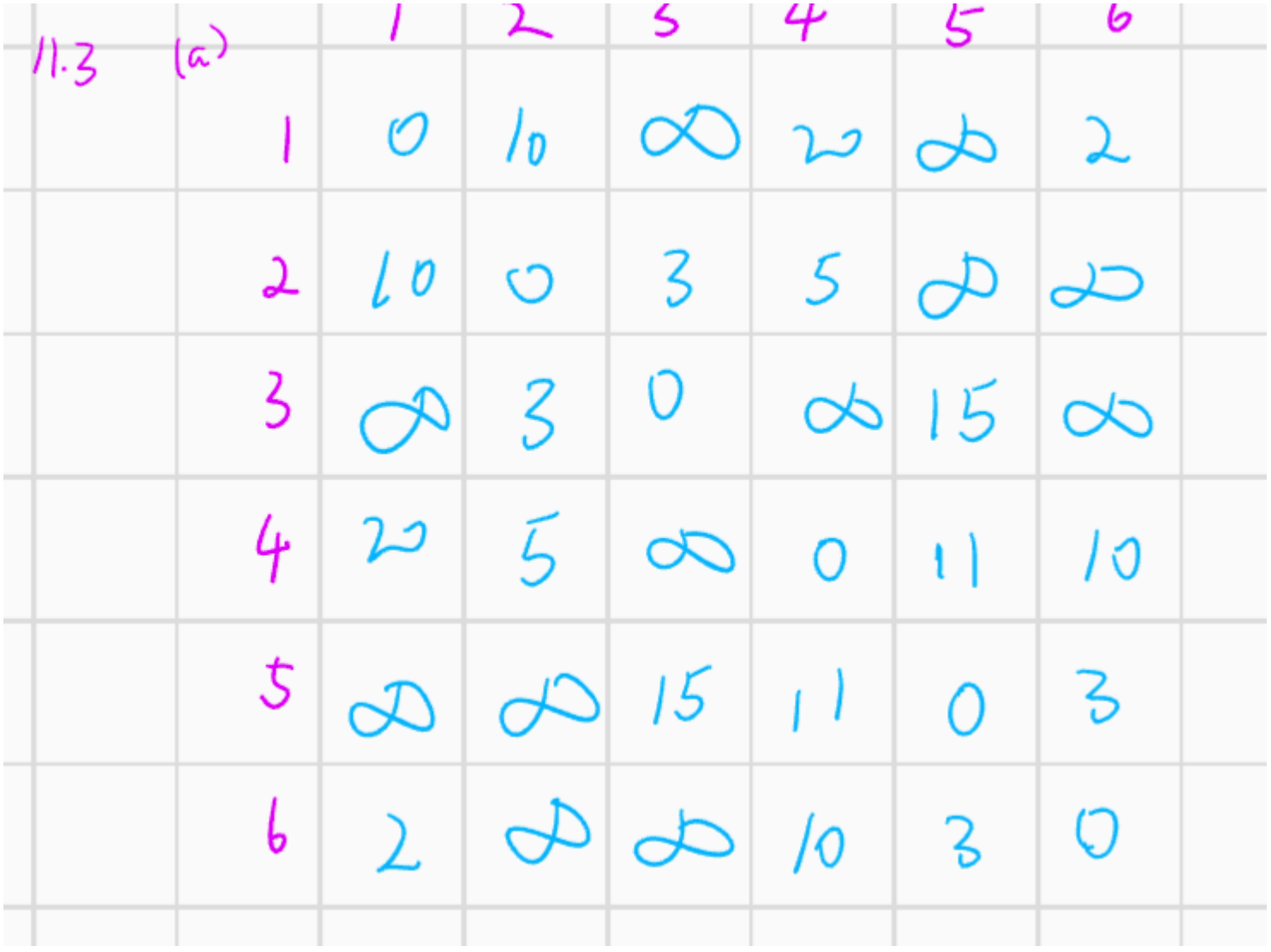
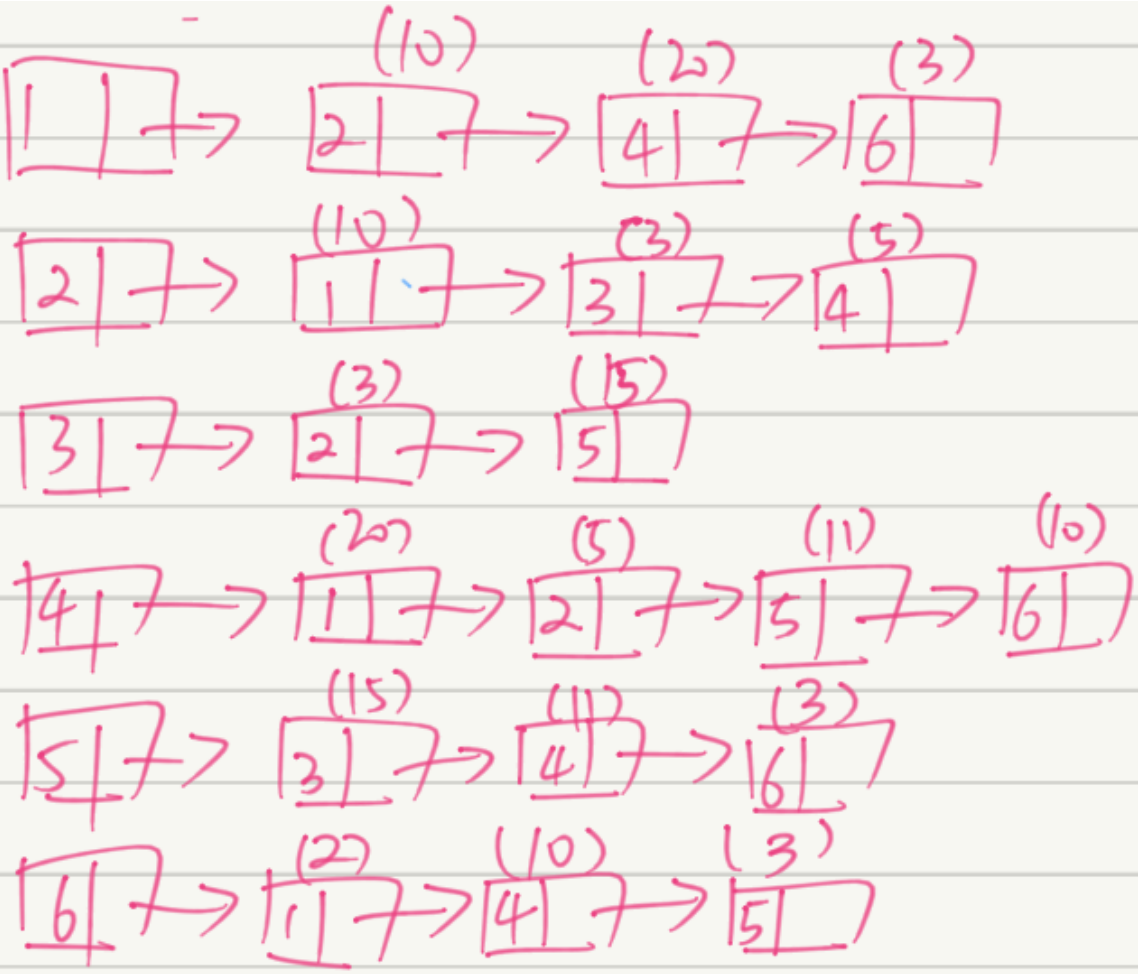
第11章 Graph

20171414930004 常家奇

11.3



(b).



(c) Matrix: $6 \times 6 \times 2 = 72$

List: $6 \times (4+2) + 18 \times (4+2+2)$

$= 6 \times 6 + 18 \times 8$

$= 180$

List

(d) Matrix: $6 \times 6 \times 1 = 36$

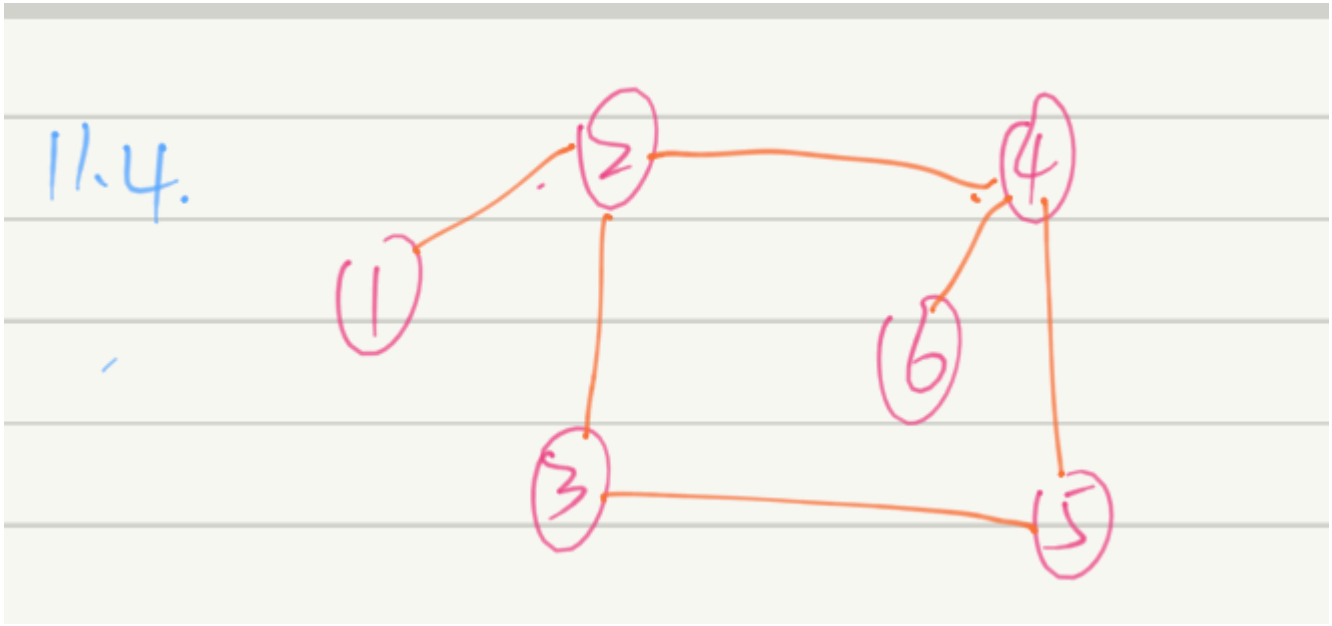
List: $6 \times (4+1) + 18 \times (4+1+2)$

$= 625 + 13 \times 7$

$= 156$

List.

11.4



11.5

```
vector<int> G[maxn];
int vis[maxn];
memset(vis,0,sizeof(vis));
void dfs(int x){
  if (vis[x]) return;
  /*
    Do sth for node x;
  */
  vis[x] = 1;
  for (int n : G[x])
    dfs(n);
}
```

11.8

增加一個訪問次數的數組，檢測到環(vis[i] > 1)就輸出。

11.15

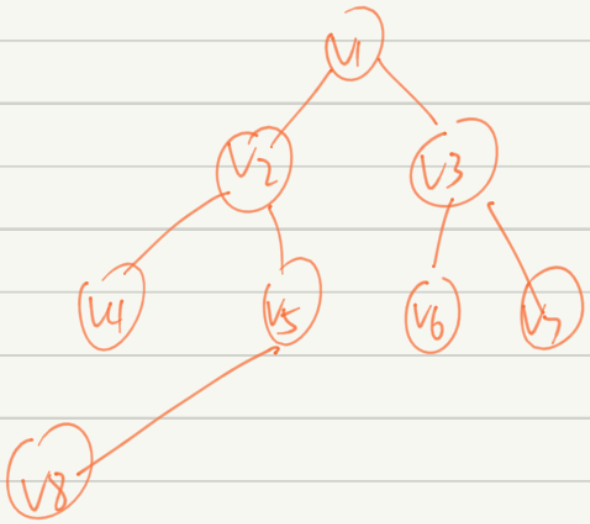
```
vector<int> graph[maxn];
int vis[maxn];
// memset(vis,0,sizeof(vis));
stack<int> res;
bool dfs(int x){
  vis[x] = -1;
  for (int next : graph[x]){
    if (vis[next] == -1) return false;
    else if (!vis[next]) dfs(next);
  }
  vis[x] = 1;
  res.push(x);
  return true;
}

/* for (int i=0;i<n;i++){
*    if (!vis[i]){
*      if (!dfs(i)) // have circle.
*    }
*  }
*
*/
```

一、 A AB

二、上 in $V_8, V_4, V_5, V_2, V_6, V_7, V_3, V_1$

(2) $V_1\ V_2\ V_3\ V_4 V_5\ V_6\ V_7\ V_8$

(3) $V_1\ V_3\ V_6\ V_7\ V_2\ V_5\ V_4\ V_8$



2. A C B D E

A C D B E

A C D E B

3

```
// struct list{
//   int v;
//   struct list* next;
// }
const int INF = 1e7;
void convert(int G[][],int n,vector<list*>& l[]){
  for (int i=0;i<n;i++)
    for (int j=0;j<n;j++) G[i][j] = INF;
  for (int i=0;i<n;i++){
    list* traval = l[i];
    while (traval){
      G[i][traval] = traval->v;
      traval = traval->next;
    }
  }
}
```

```cpp
vector<vector<int>> graph;
const int maxn = 10000 + 10;
int n; // count of v
int in[maxn];
int out[maxn];

void count(vector<vector<int>>& graph){
  memset(in,0,sizeof(in));
  memset(out,0,sizeof(out));
  for (int i=0;i<graph.size();i++){
    out[i] = graph[i].size();
    for (int x : graph[i])
      in[x] += 1;
  }
}

vector<int> get_root(vector<vector<int>>& graph){
  vector<int> res;
  count(graph);
  for (int i=0;i<n;i++)
    if (!in[i]) res.push(i);
  return res;
}
```

4. (1) i: 0→n

j: i+1→n

if (graph[i][j]) cnt++;

(2) 看 graph[i][j] 是否等于 INF.

(3) 入度+出度

5. d

A	B	C	D	E
0	10	17	15	17

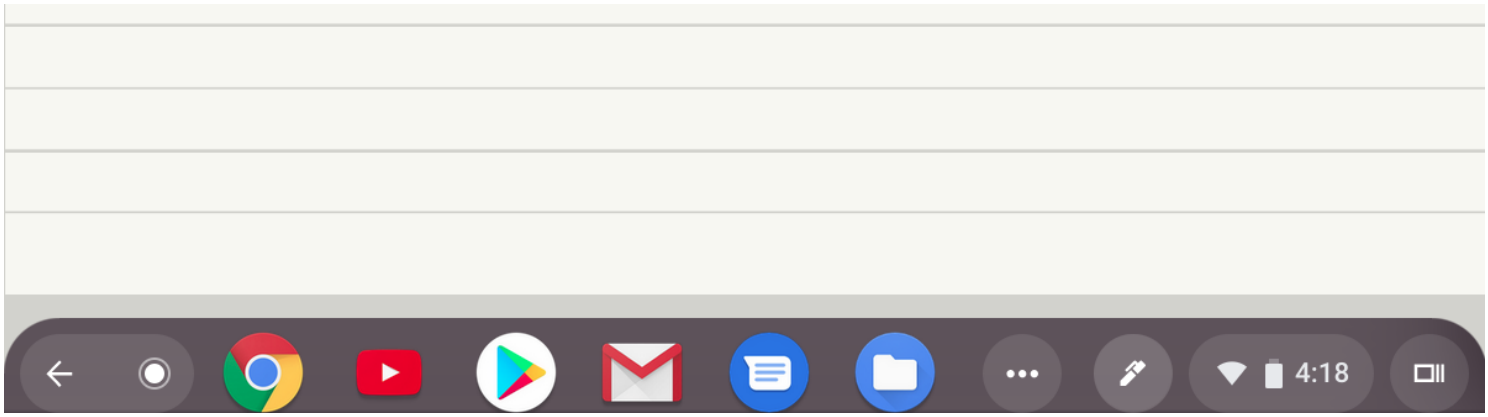
prev	A	B	C	D	E
	A	A	D	B	D

A→B: A, B (10)

A→C: A B D C (17)

A→D A B D (15)

A→E A B D E (17)


4:18