

i	1	2	3	4	5	6	7	8	9
C_i	1 ²	2 ²	1	4 ²	1	1	1	8 ²	1

$$\sum_{j=1}^n c_j = (n - \lfloor \log_2 n \rfloor) + \sum_{j=1}^{\lfloor \log_2 n \rfloor} (2^j)^2$$

$$\leq n + \sum_{j=1}^{\log_2 n} 4^j$$

$$= n + \frac{4(4^{\log_2 n} - 1)}{4 - 1}$$

$$= \frac{4}{3}n^2 + n$$

$$= O(n^2)$$

②

Valid topological sorts:

1. (a) (b) (c) (d) (e) (f)
2. (a) (b) (d) (c) (e) (f)
3. (a) (b) (d) (e) (c) (f)
4. (a) (d) (e) (b) (c) (f)
5. (a) (d) (b) (c) (e) (f)
6. (a) (d) (b) (e) (c) (f)

③

To detect a cycle in undirected graph we will modify DFS a little:

Modified-DFS (G) {

for each vertex $u \in G$
color [u] = white;
parent [u] = nil;

for each vertex $u \in G$
if color [u] = white
DFS-VISIT (u);

}

DFS-VISIT (u) {

color [u] = gray; cycleExists = false;

for each $v \in \text{adj}[u]$

if (color [v] = white)

then parent [v] = u ;

cycleExists = DFS-VISIT (v);

else if (color [v] = gray and
parent [u] $\neq v$)

return true;

color [u] = black;

return cycleExists;

}

This is the same algorithm as DFS and has running time of $(n+m)$ where n = number of vertices and m = number of edges.