WEEK4:

Char 1 byte

Int, float 4 bytes

Double 8 bytes

Any\_type \* 4 bytes

scanf("%d", &numberOfElems);

int i, \*vector;

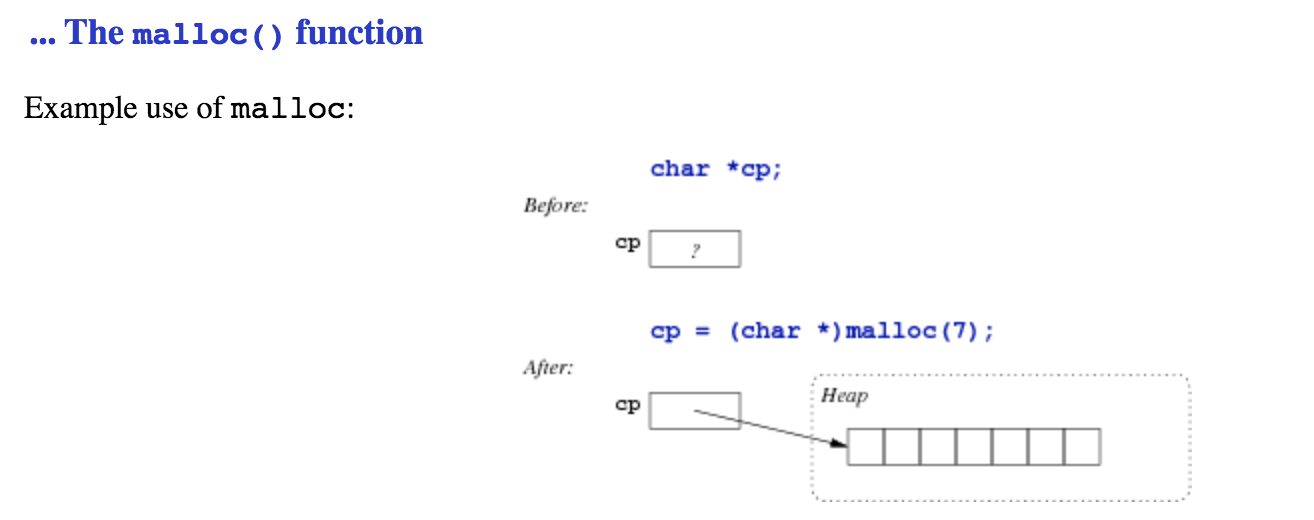
size\_t numberOfBytes;

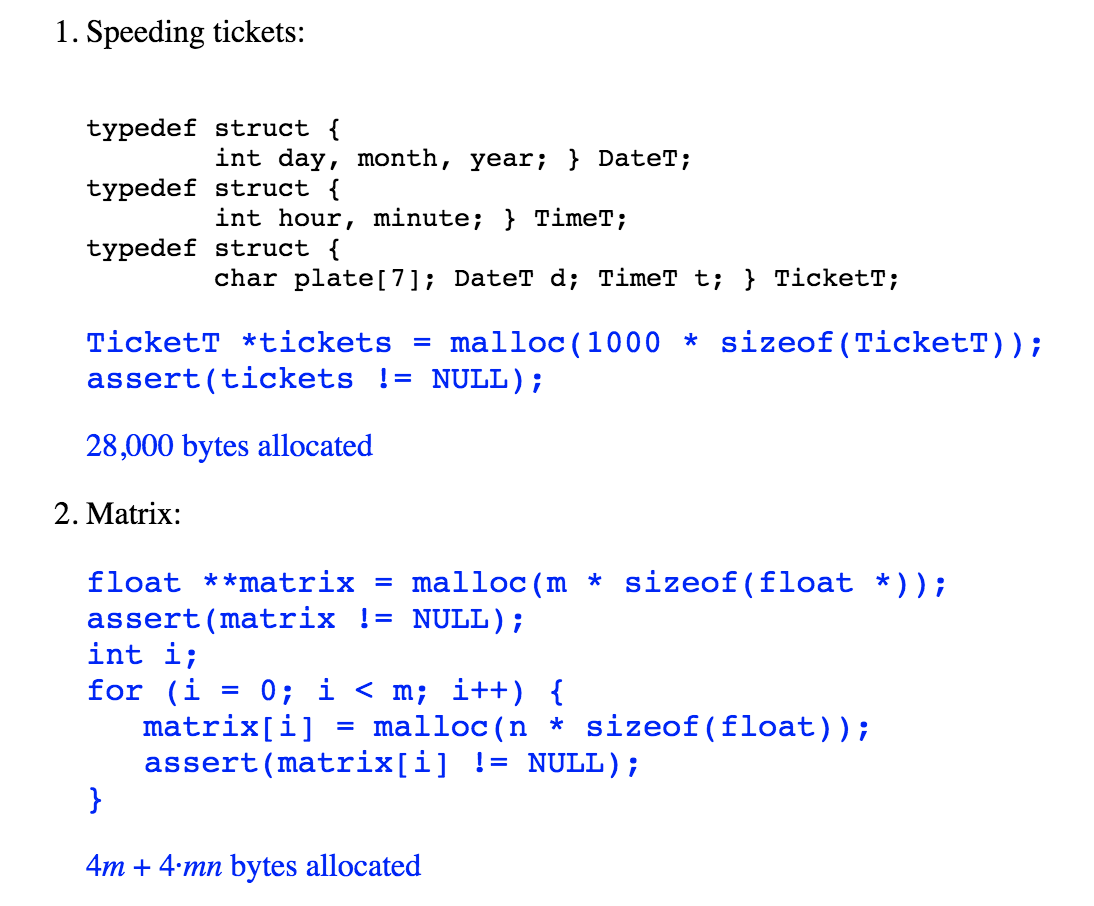
numberOfBytes = numberOfElems \* sizeof(int);

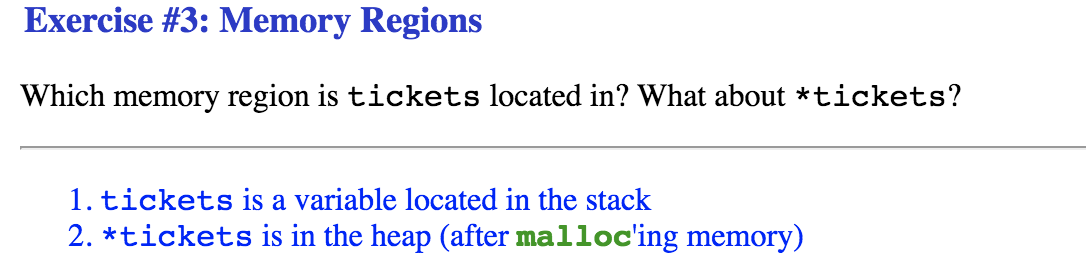
vector = malloc(numberOfBytes);

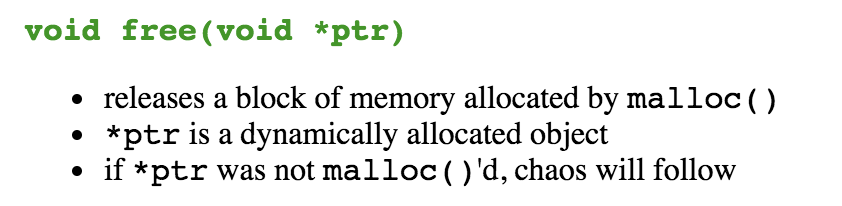
The malloc() function:

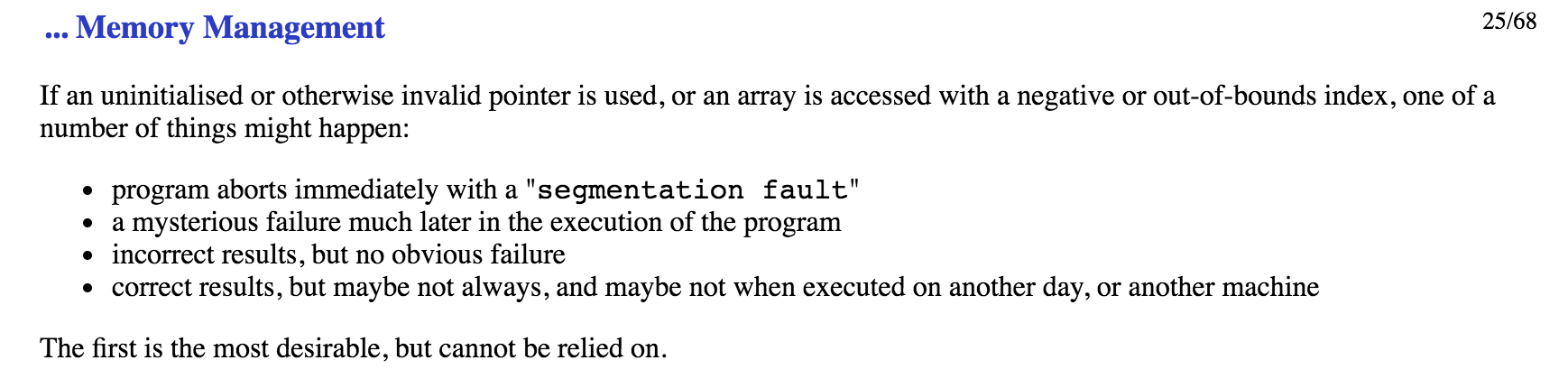
attempts to reserve a block of n bytes in the heap

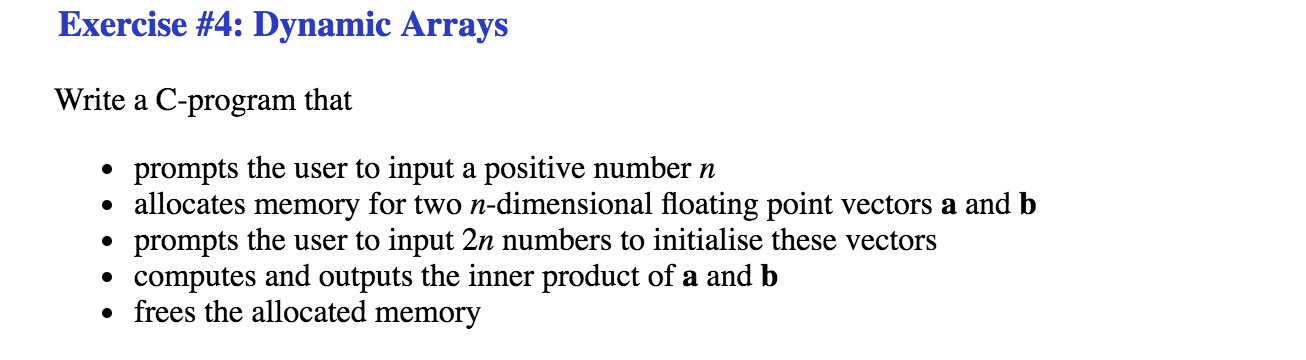


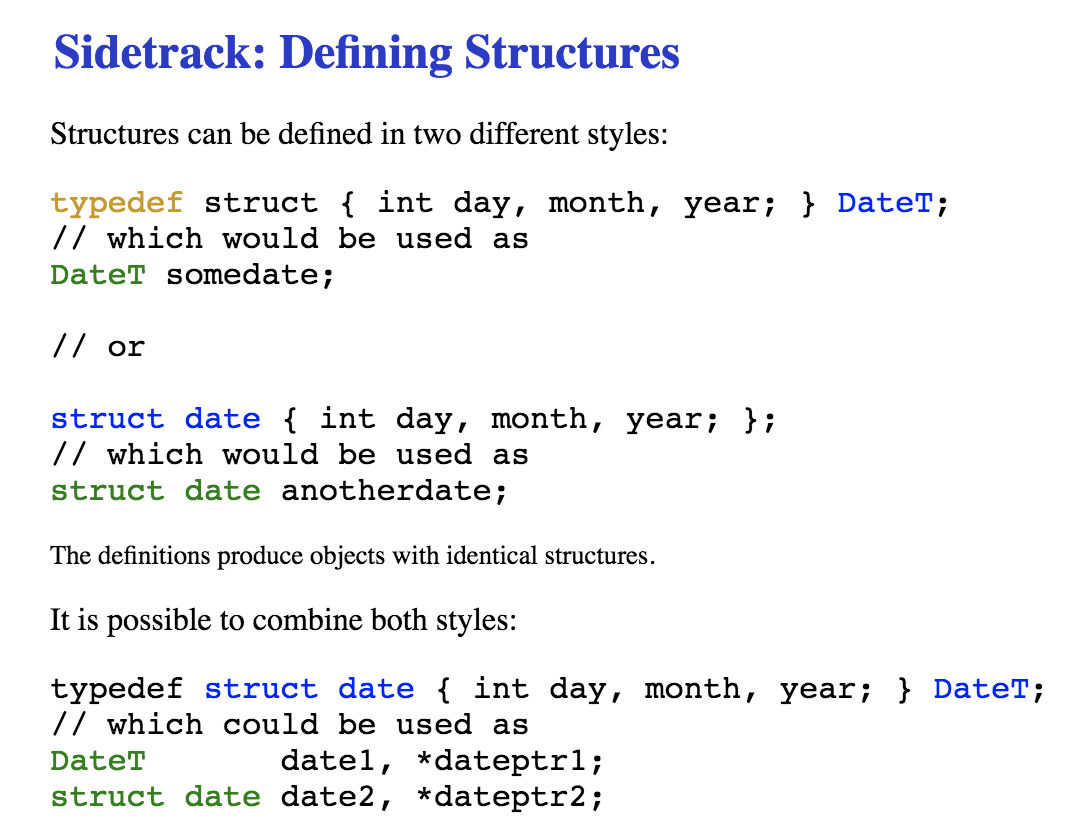


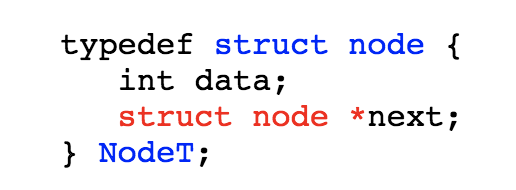


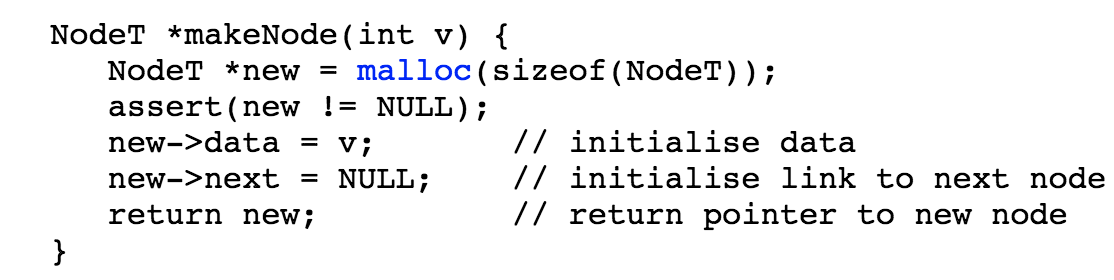


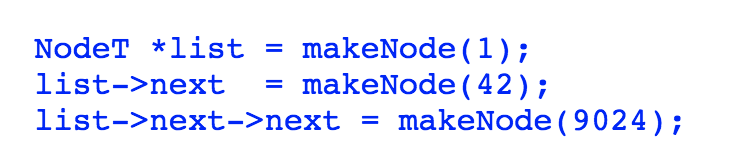


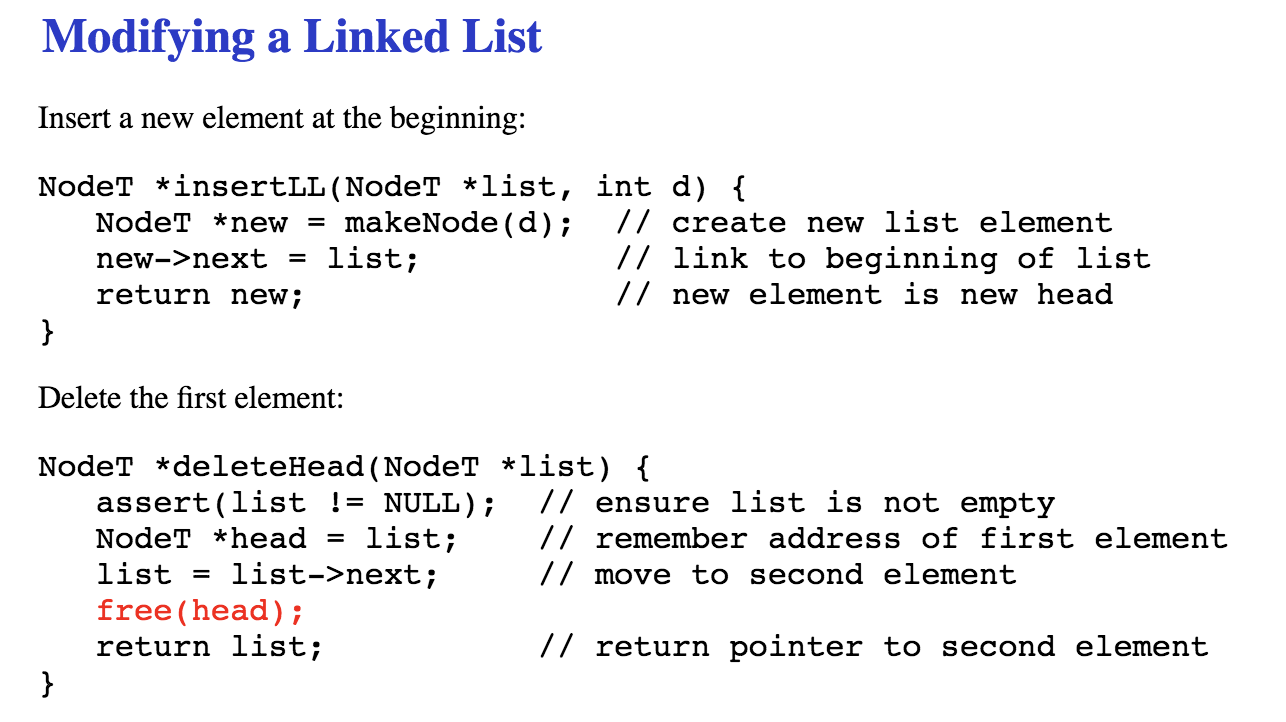


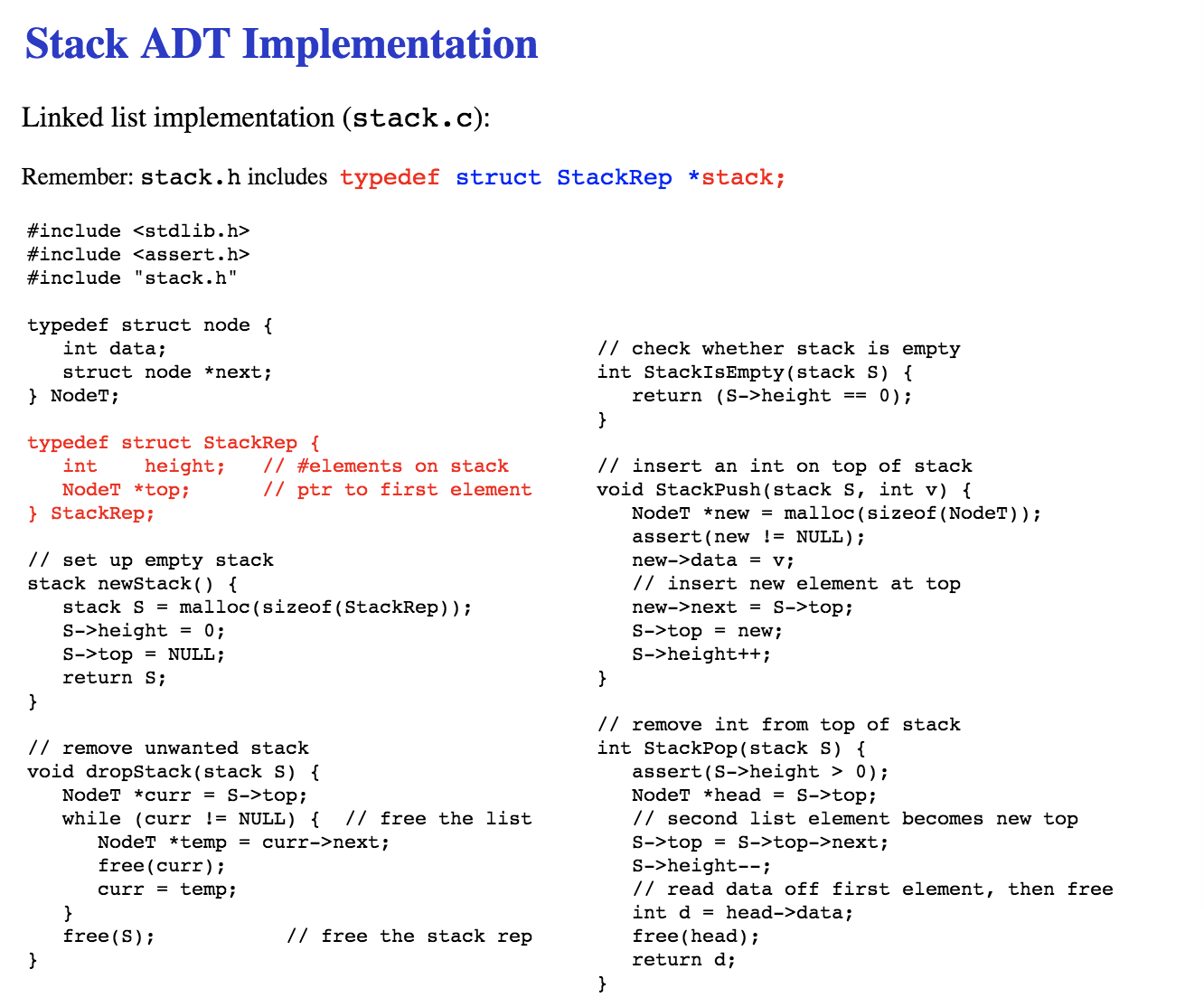




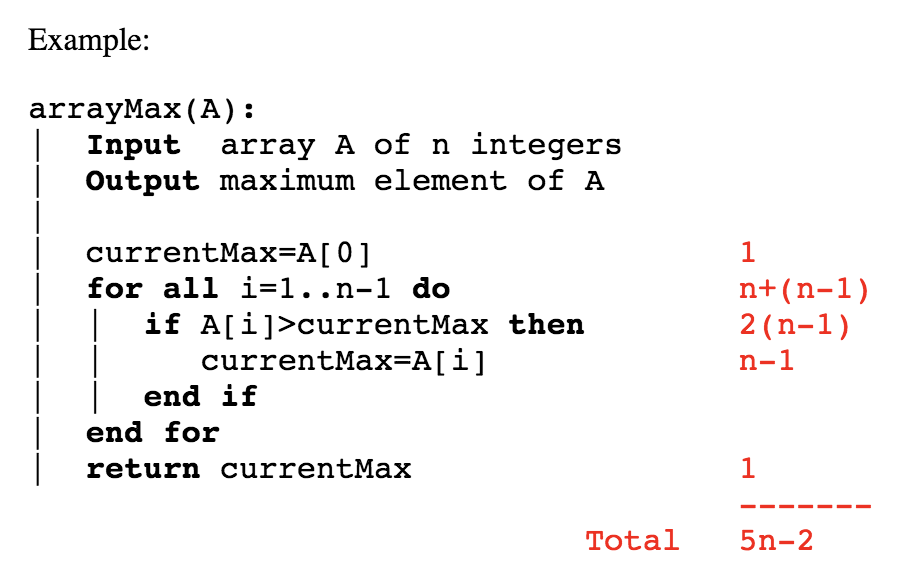




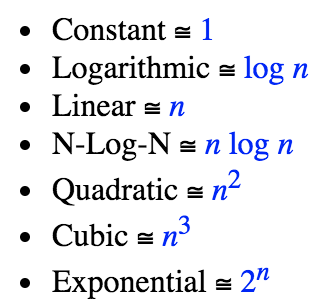


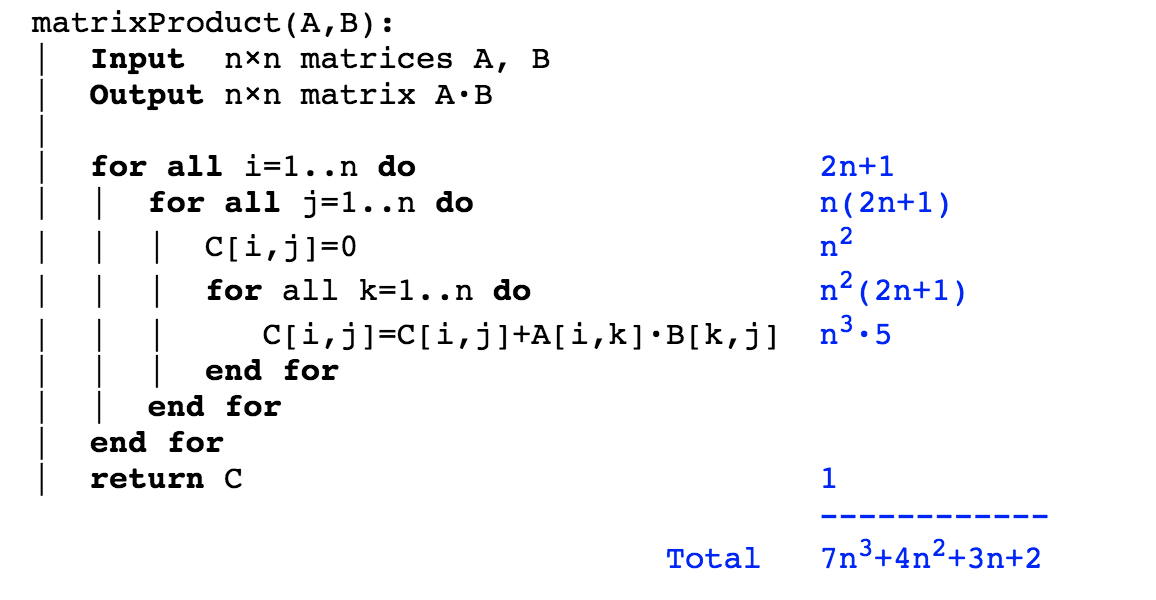


WEEK5:

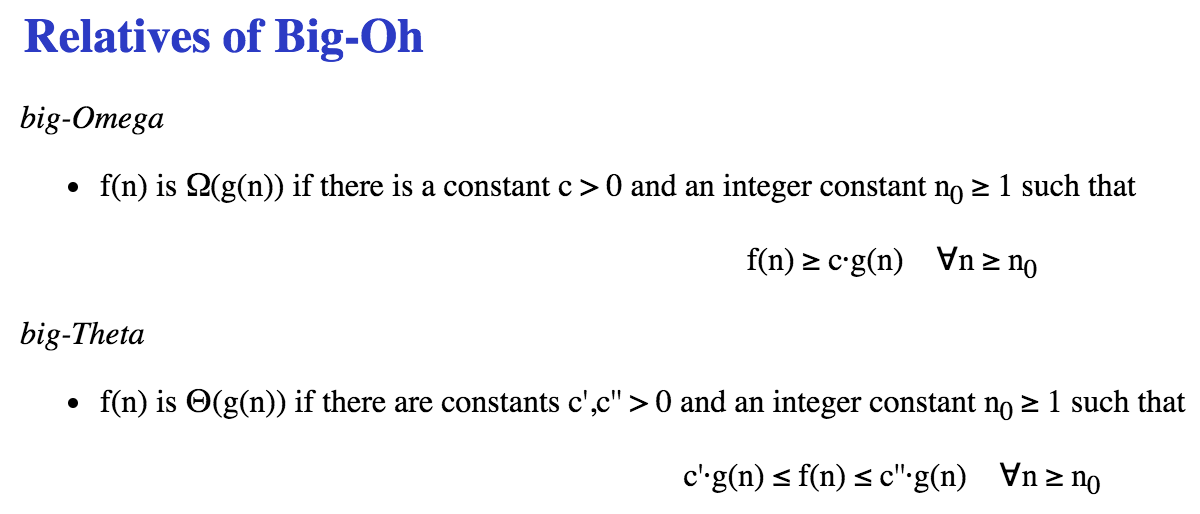


？？？？？

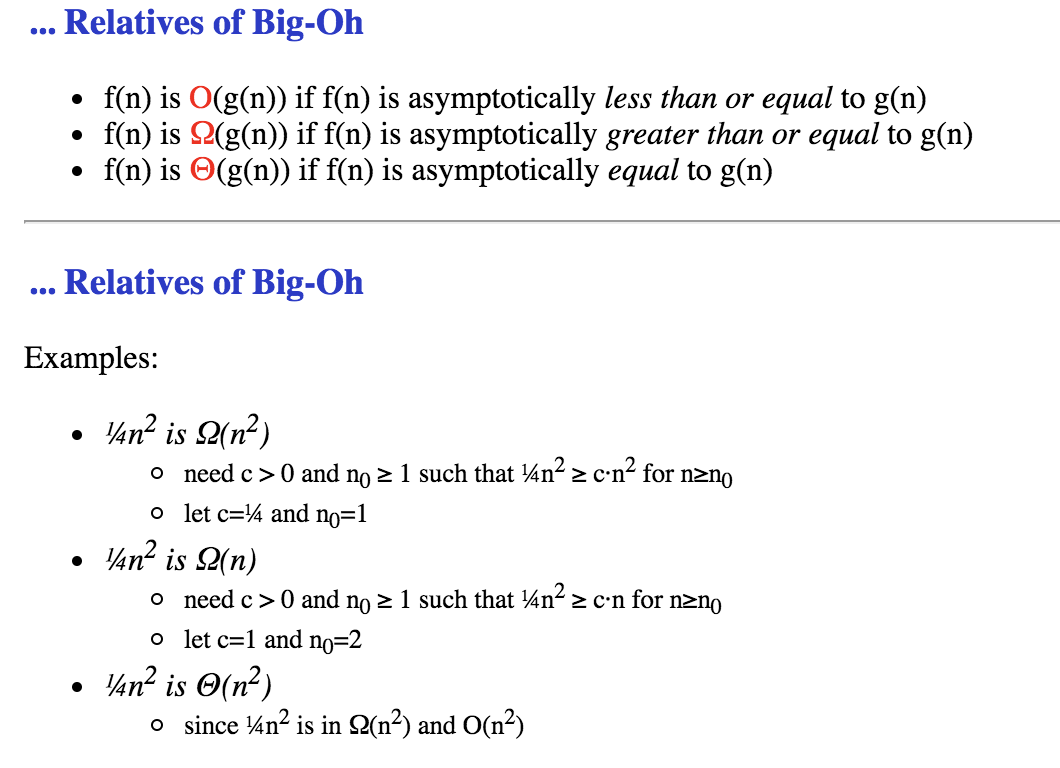


？？？？

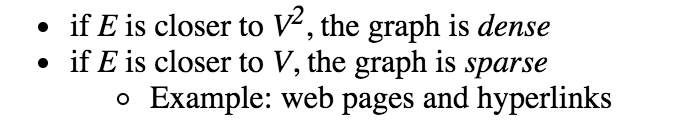
Estimating Running Times、Determine the number of primitive operations

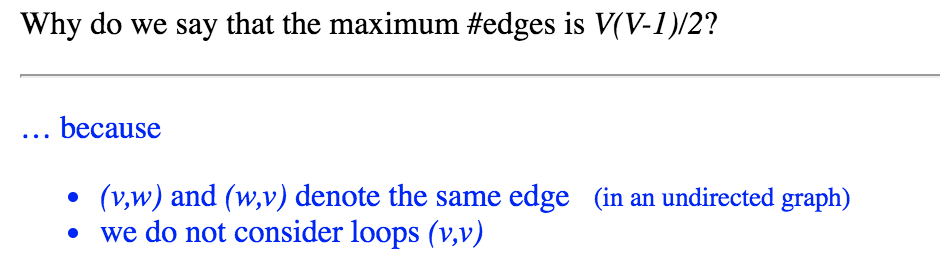


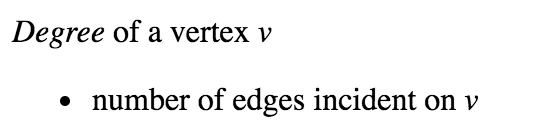
prefix averages:

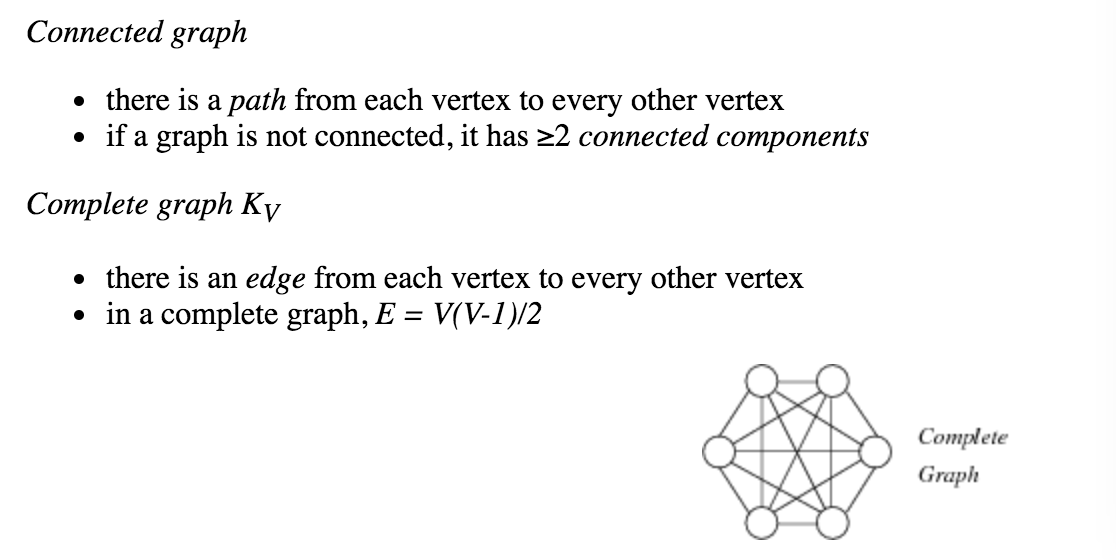


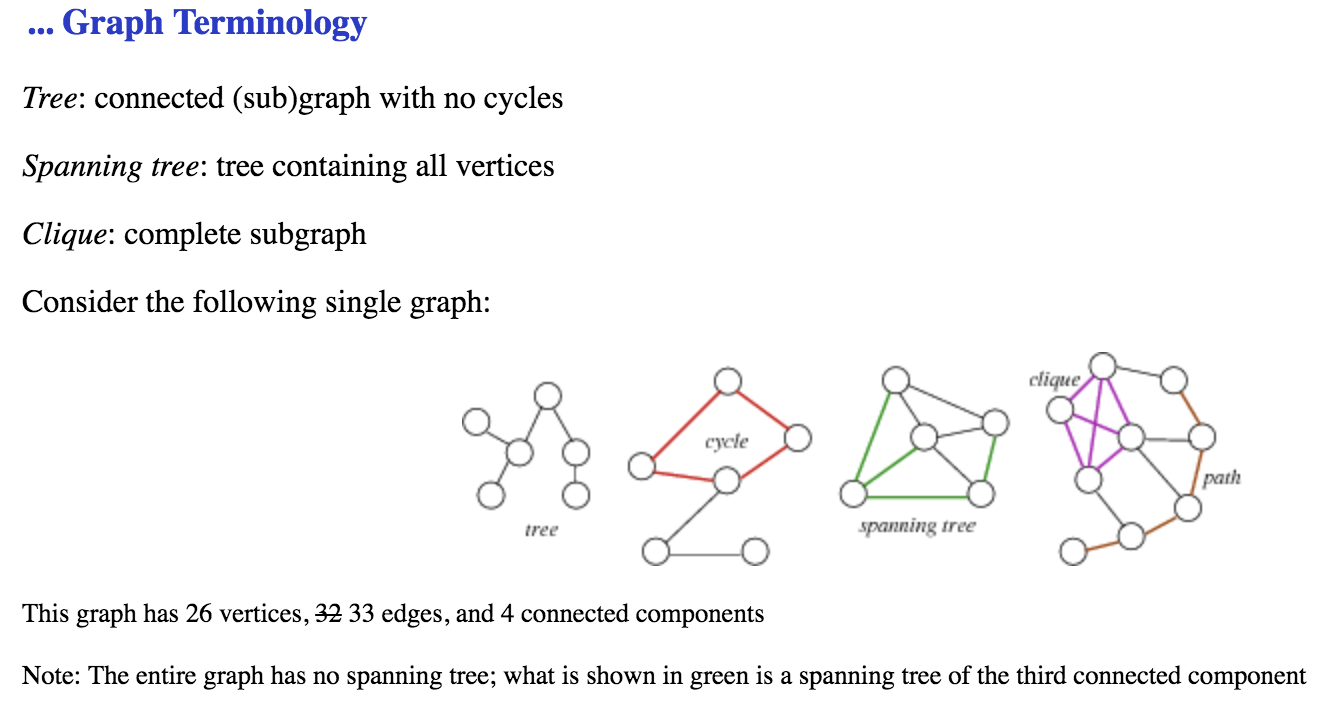
WEEK6:

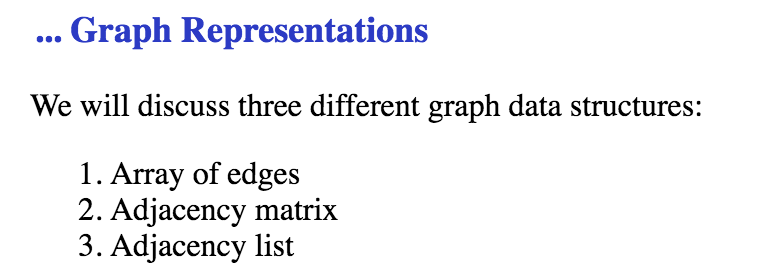


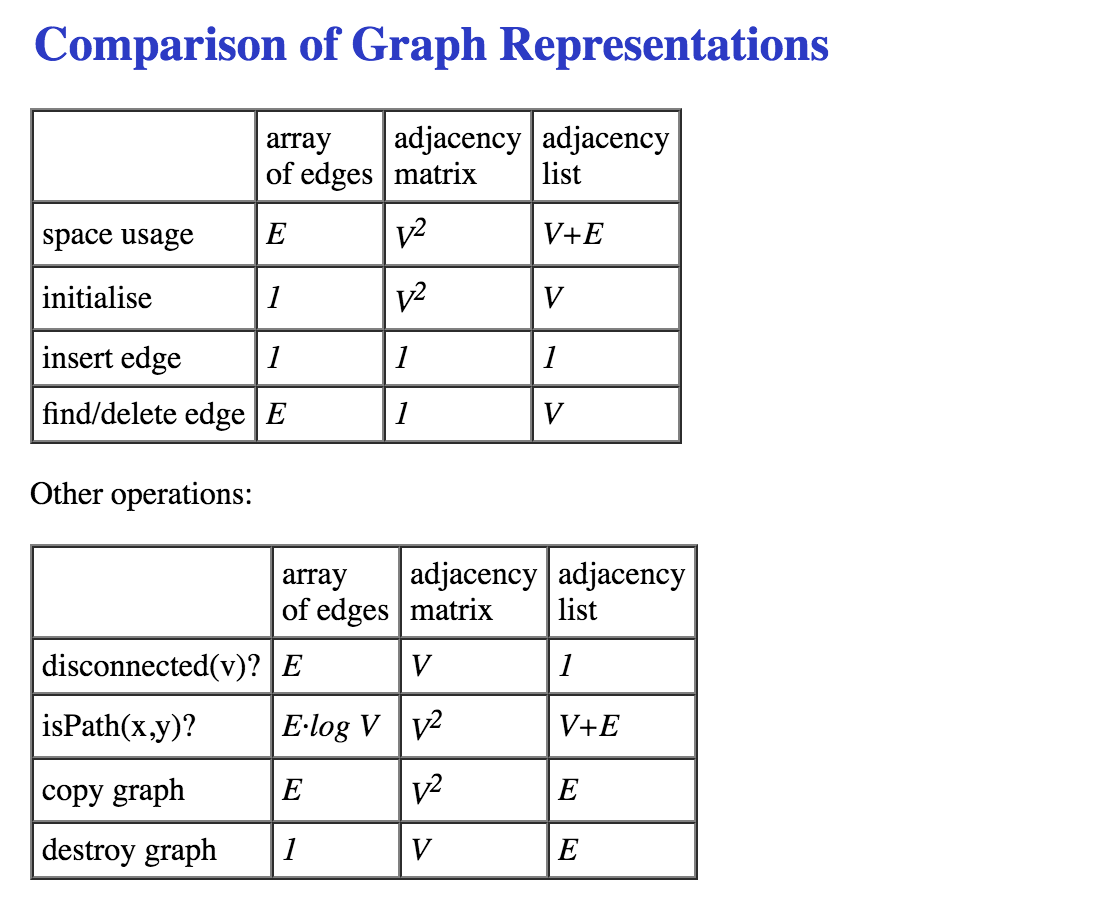


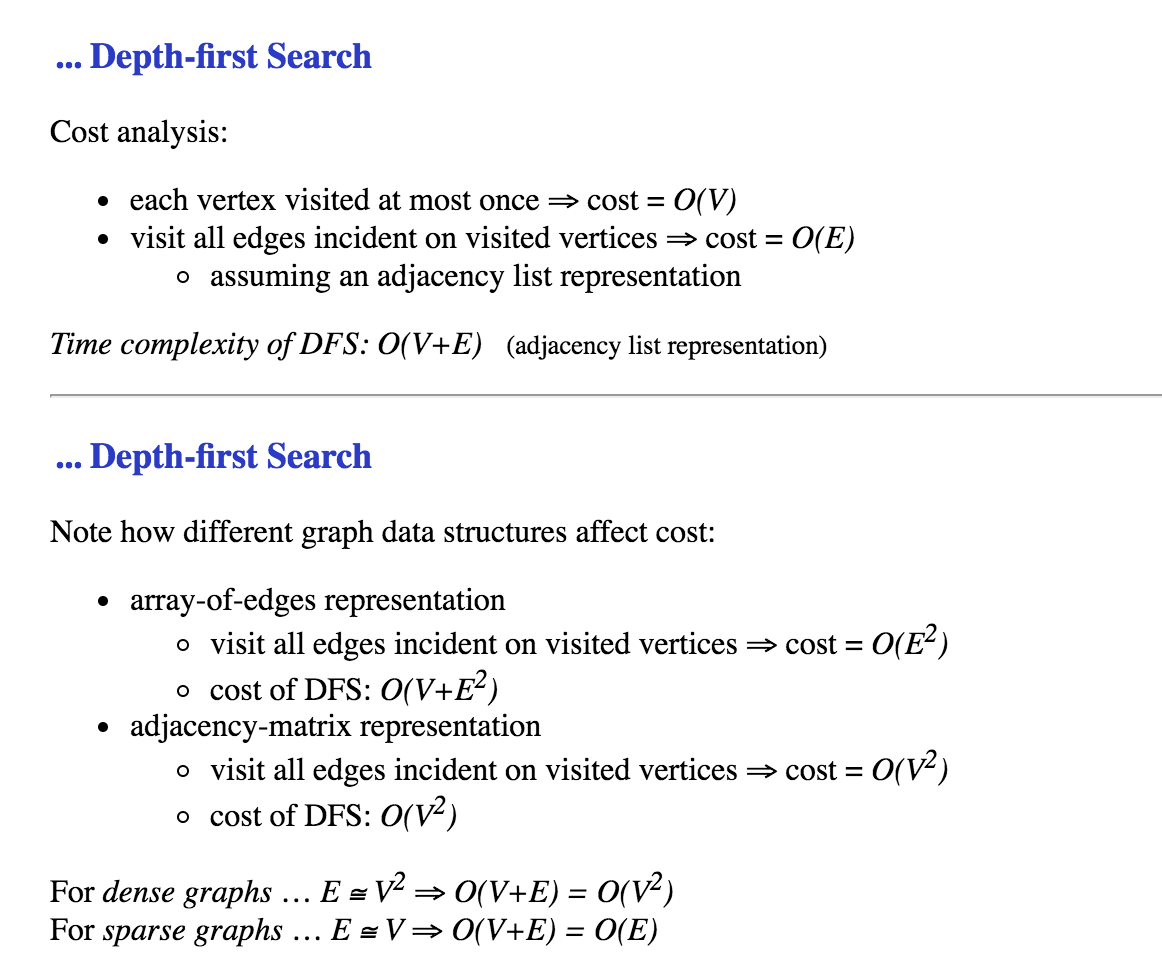




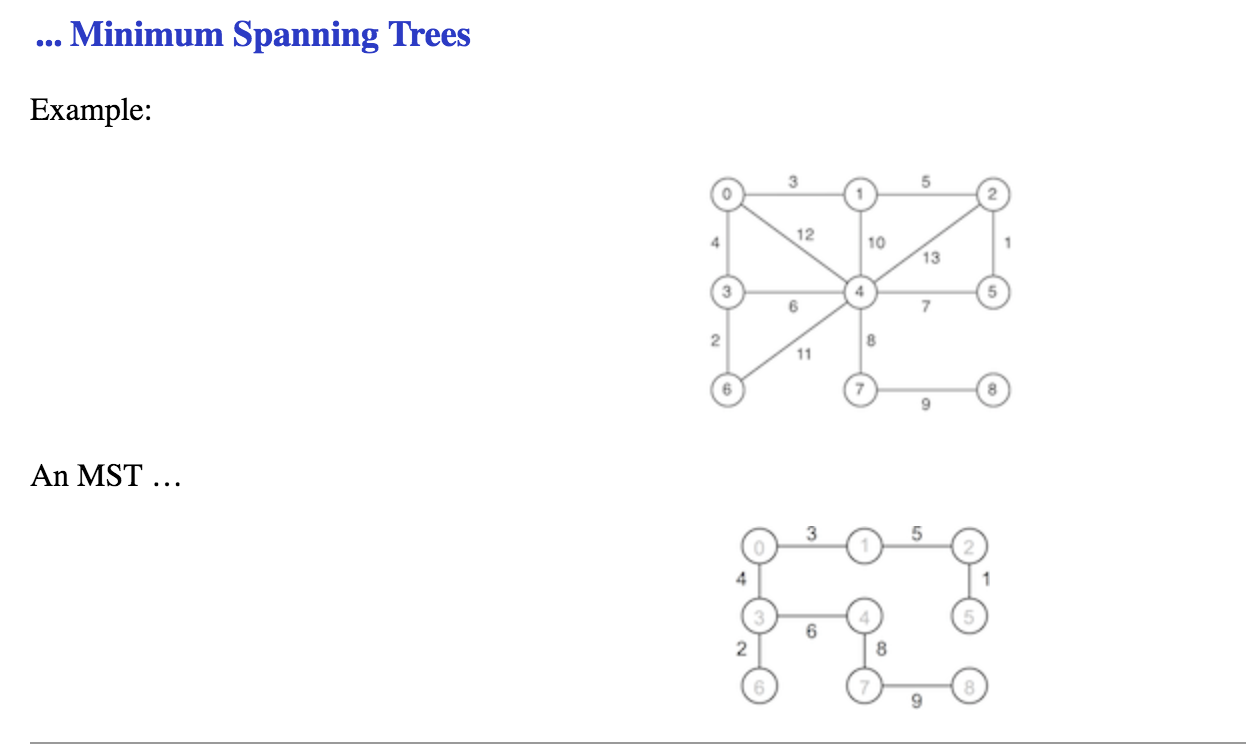




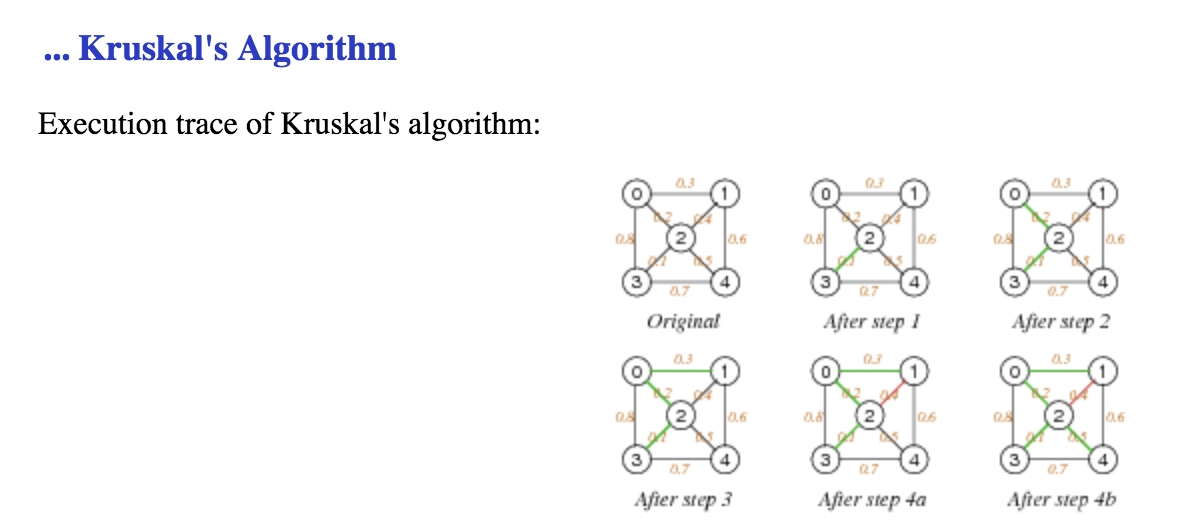


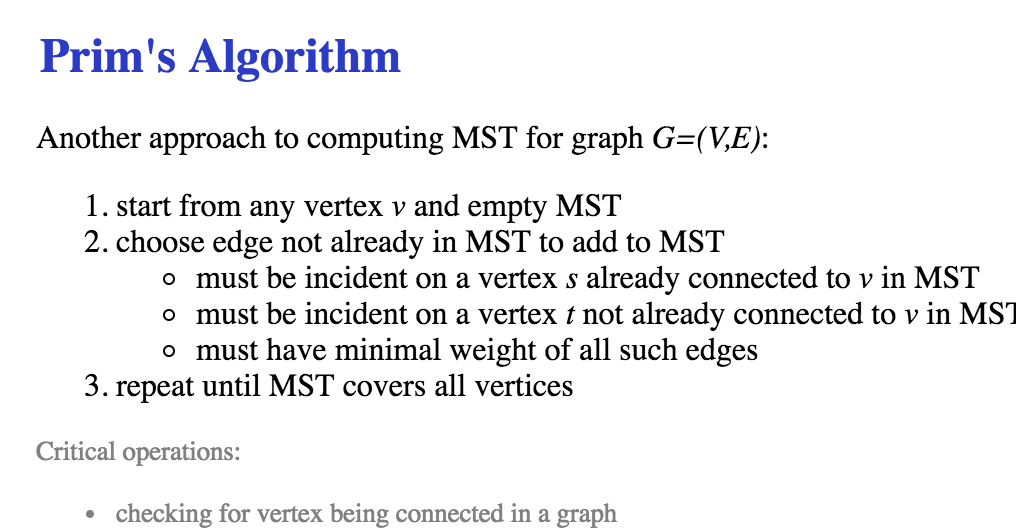


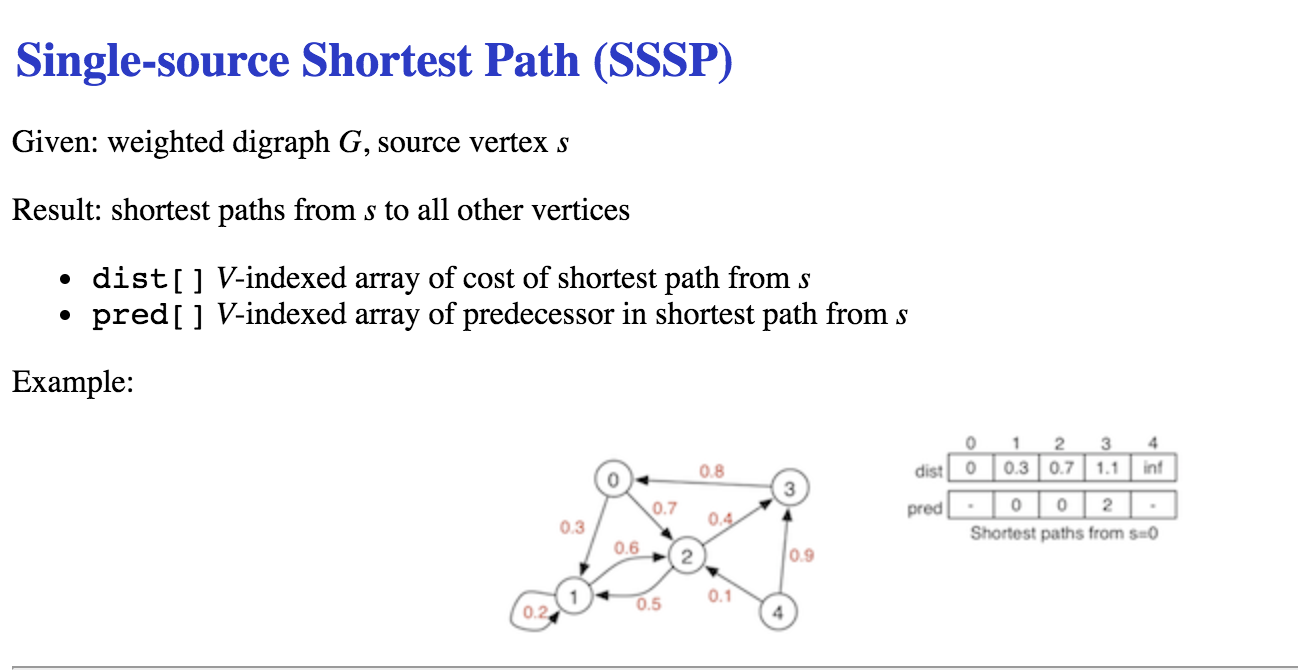
WEEK8:

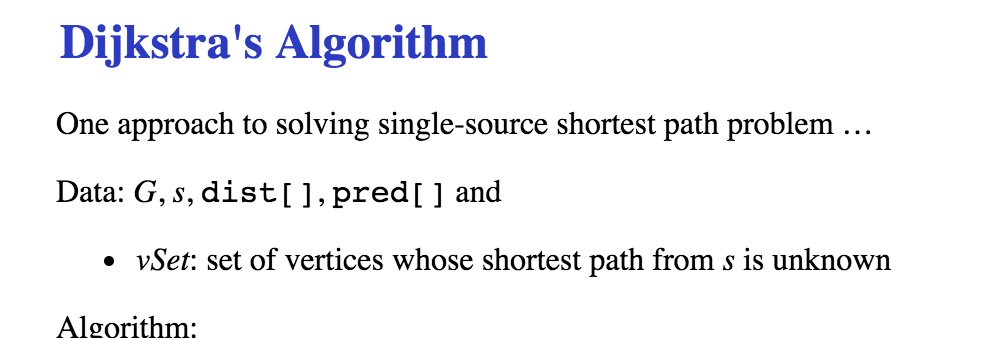


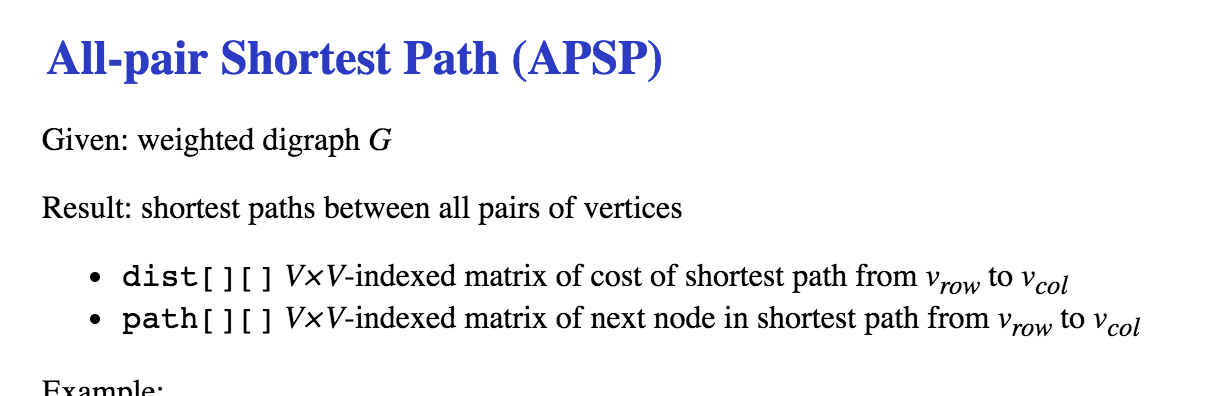
find minimum spanning tree

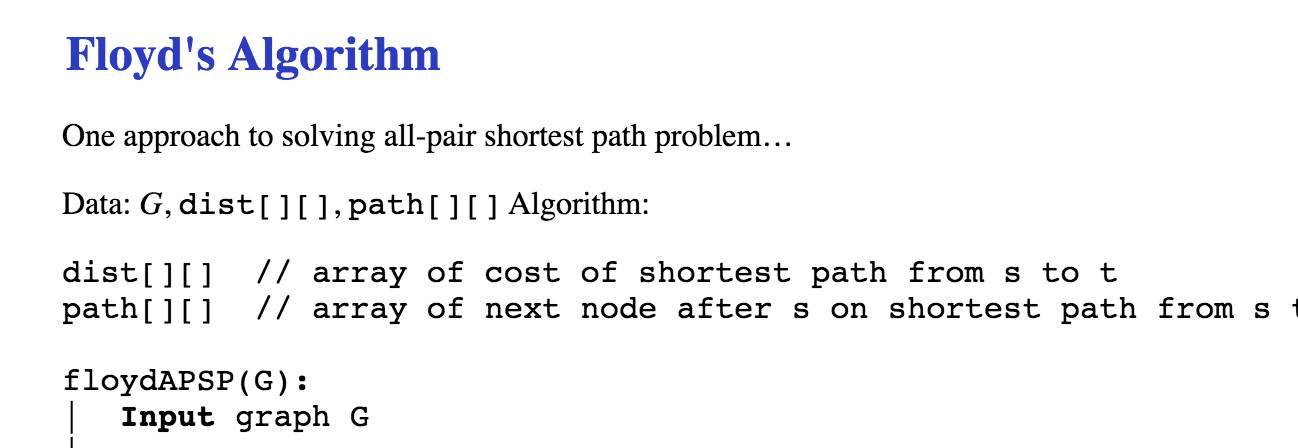


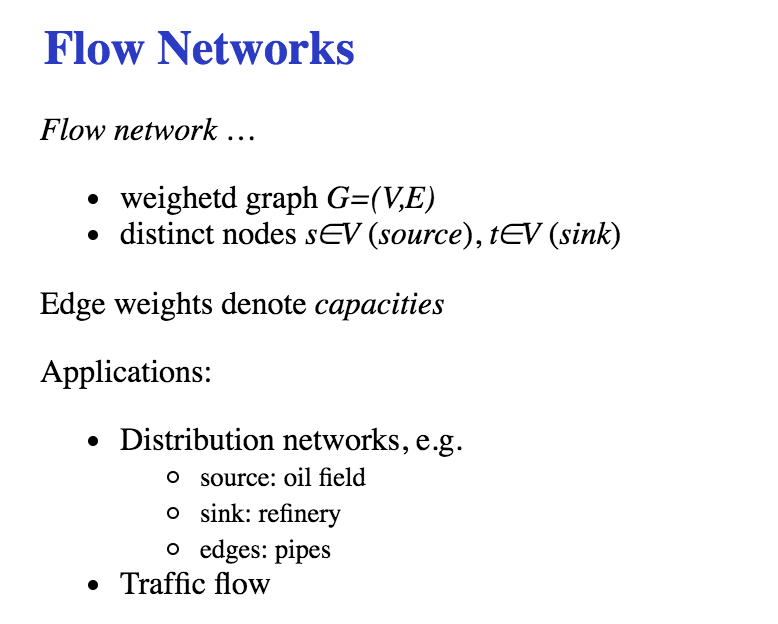












Residual Network

Edmonds-Karp Algorithm

**Edmonds-Karp Algorithm**

|  |  |
| --- | --- |
| **Summary** | 61/61 |

* Weighted graph representations
* Minimum Spanning Tree (MST)
  + Kruskal, Prim
* Shortest path problems
  + Dijkstra (single source SPP)
  + Floyd (all-pair SSP)
* Flow networks
  + Edmonds-Karp (maximum flow)

* Suggested reading (Sedgewick):
  + MST … Ch.20-20.4
  + SSP … Ch.21-21.3
  + Flow … Ch.22.1-22.2

WEEK 9 :

Searching 搜索算法：

