思考點：

1. 各 SD request 的 path
2. 選取哪些 SD request
3. 各節點的 memory 上限
4. 各 SD request 的 tree 要長怎樣
5. ~~是否要延時（拉長 tree 高度）~~
6. ~~是否要繞路~~
7. BFS
8. 從path短的request開始選
9. 若有memory滿則跳過
10. 靠左Complete→靠左Complete→S>D linear→D>S linear \* n為S→D距離
11. Complete 高度：floor( log2(n-1) ) + 3 ； linear 高度：n+1
12. 若linear可放，判斷(去掉最上排)四角的2\*2格中最小memory剩餘
13. 選三者中最大(以Z字編號四角)，若非0：34 > 3大左C, 3小右C、14 > DS、23 > SD；若相同，依4. 順序。 →紀錄t與圖形 / 更新memory
14. 若選完不行試 t+1，直到 剩餘t < complete高度

//4 input

//memory input

// + make memory\_count array

//link input

// + make graph

//request input

// + BFS

// + save 5 SD\_path array & node count & two type tree height

// + sort(heap) (using node count) & save it

//while(time is enough for complete\_tree)

// if(linear available) compare 4 corner

//  + make\_tree & personal time bar

// if(memory is enough) update memory\_count array

//  + save accept\_request & break;

// if(request is not accepted)

// + change path & push it back to heap

//print ???