1.

(1) 숫자 9개 제시해주고 bubble sort와 recursive quick sort의 과정을 보여라.

(2) Bubble sort와 merge sort에 소요되는 시간(Big O)과 이유

2. weight 나타난 그래프 줌

(1) weight가 무거운 순서대로 depth-first search 탐색 순서 나타내고 DFS 트리 그리기

(2) Prim’s 방법으로 spanning tree 그리기

3. 해시

(1) Uniformed hash function 공식 쓰기

(2) linear probing 공식

(3) linear probing 단점

4. (1) AVL 트리를 사용하는 목적

(2) 숫자 제시하고 한 개씩 추가해서 AVL 트리 만드는 과정 나타내기 (추가한 노드와 바로 위 노드의 balance factor 표시해야함)

5 . (1) 노드가 3개인 그래프 주고 3x3의 비용행렬 A-1, A0, A1, A2 채우기

(2) 노드가 4개인 그래프 주고 4x4인 A(인접행렬), A+(이행적 폐쇄행렬), A\*(반사 이행적 폐쇄행렬) 채우기

6. (1) 작업 네트워크를 사용하는 이유, 임계도의 의미

(2) 임계경로 나타내어라. 임계작업을 나타내고 의미를 써라

(3) Earliest time, Iatest time, 임계도 표 채우기

7. 힙소트 코드

void HeapSort(element list[], int n)

{

Int I, j;

element temp;

for ( ; i>0; i--)

adjust(list, , );

for ( ; i>0; i--) {

SWAP(list[1], list[i+1], temp);

adjust(list, , );

}

}

Void adjust(element list[], int root, int n) {

Int child, rootkey;

Element temp=list[root];

rootkey= ;

child=2\*root;

while (child<=n) {

if ((child<n) && list[child].key<list[child+1].key) child++;

if ( ) break;

else {

= ;

child\*=2;

}

}

list[ ]= temp;

}

사실 빈 칸이 어디였는지는 헷갈리는데 ㅜㅜ 힙소트 코드가 나왔어요.

수업시간에 강조해주신 부분에서 잘 나옵니다. 문제도 단원 별로 균등하게 분배되어 있습니다.