작성자 : 시대컴퓨터학원

# 알고리즘 정리

☐ 알고리즘 정의와 요건

1. 입출력 : 0개 이상의 외부 입력과 하나 이상의 출력이 있어야 한다.
2. 명확성 : 각 명령은 모호하지 않고 단순 명확해야 한다.
3. 유한성 : 한정된 수의 단계를 거친 후에는 반드시 종료해야 한다.
4. 유효성 : 모든 명령들은 실행 가능해야 한다.
5. 효율성 : 실용적인 측면에서 보는 것이다.(추가내용임)

☐ 자료구조 종류/정의와 구분

1. 선형구조 : 배열, 연결리스트, 큐, 스택
   - 배열 : 같은 성질의 자료를 기억, 자료형+배열명+(인덱스,원소)로 구성, 연속적으로 메모리에 할당 됨, 논리적 물리적 순서가 같음으로 접근이 빠르다. 순서를 유지해야 함으로 매번 자료의 이동이 필요한 단점이 있어 연결리스트로 구현 함.
   - 연결리스트 : 데이터+링크+주소
   - 큐 : FIFO , 스택 : LIFO

2. 비선형구조 : 비-트-그(트리, 그래프)
   - 트리 : 하나 이상의 노드로 구성된 구조,
   - 이진트리의 종류 : 전 이진트리(모든 노드의 차수가 0이거나 2인 이진트리), 포화이진트리(모든 잎의 경로가 같은 전 이진트리), 완전 이진트리(위에서 아래로, 왼쪽에서 오른쪽으로 차있는 트리), 균형 이진트리(모든 단말노드의 깊이 차이가 많아야 1인 트리)

☐ 그래프 용어

- 두 정점이 인접되어 있다, 두 정점이 부속되어 있는 것이 특징
- 무방향 그래프 : {(1,2), (1,3)...} ( )로 표현
- 방향그래프 : {<1,2>, <1,3>...} < >로 표현
- 인접, 부속 : 두 정점 사이에 간선이 있으면 u와 v는 이접, 해당 간선은 점점 u와 vdp 부속
- 부분 그래프 : V(G') ≤ V(G)이고 E(G')≤E(G)인 그래프 G'를 G의 부분 그래프라 한다. 즉 큰 쪽에 속해 있다는 뜻
- 경로 : 간선으로 연결 된 정점의 순서 리스트, 이때 경로에 존재하는 간선의 개수를 경로의 길이
- 차수 : 정점에 부속된 간선의 수, 진입차수는 들어오는 것←, 진출차수는 나가는 것→
- 단순경로 : 처음과 마지막 정점을 제외한 모든 정점이 서로 다른 경로
- 사이클 : 처음과 마지막 정점이 같은 단순 경로, 사이클이면서 경로가 1인 것을 루프(loop)라 함
- 강력연결 : 정점 u와 v의 모든 쌍에 대하여 u에서 v로의 방향경로와 v에서 u로 방향 경로가 존재

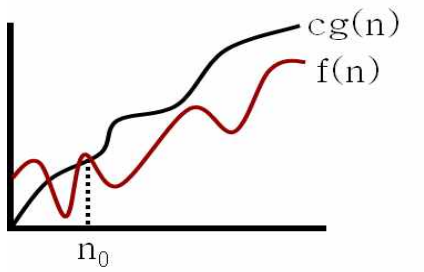
☐ 알고리즘 설계 기법과 적용된 알고리즘

- 분할정복 : 퀵, 합병, 이진탐색
- 욕심쟁이 : 거스름돈, 배낭, 크루스칼, 프림, 다이스트라, 각 단계에서 선택한 최적해들이 전체적인 최적해를 구성한다.
- 동적프로그래밍 : 플로이드, 스트링매칭, 연쇄행렬

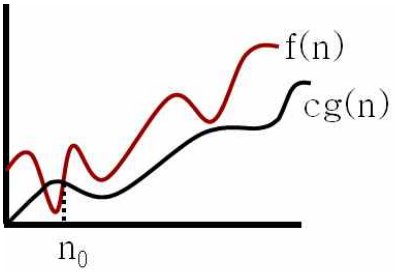
☐ 시간복잡도 관련개념

- 컴퓨터 성능이나 언어의 특성에 의존하지 않고 수행 횟수를 계산하여 효율성 분석 함
- 단위 연산의 크기보다는 입력크기 N의 함수로 표현 함
- 입력상태의 상황에 따라 시간 복잡도에 영향을 받음
- 시간복잡도는 최선이나 평균적으로 하지 않고 최악의 경우로 계산한다.
- 최고차항만 이용하고 나머지는 제거한다.
- $O(1)\rightarrow O(\log n)\rightarrow O(n)\rightarrow O(n\log n)\rightarrow O(n^2)\rightarrow O(n^3)\rightarrow O(2^n)\rightarrow O(n!)$

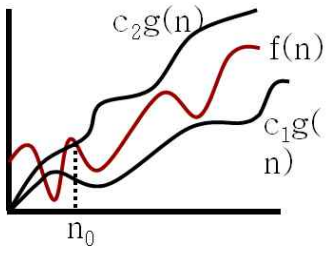
☐ O-표기법




<점근적 상한>
f(n≤c.g(n)이면 f(n)=O(g(n))이다
<참고> f(n)보다 O가 위에 있다




<점근적 하한>
f(n)≥c.g(n)이면 f(n)=Ω(g(n))이다
<참고> f(n)보다 Ω가 아래에 있다




<점근적 상/하한>
c1.g(n)≤f(n)≤c2.g(n)이면
f(n)=Ø(g(n))이다
<참고> f(n)이 센타(세타)에 있다

☐ 점화식과 폐쇄형

- $T(n) = T(n-1)+O(n)$ → $T(n) = O(n^2)$ : 최악의 경우 이진, 퀵 정렬, 평균적인 경우 삽입정렬
- $T(n) = T(n/2)+O(1)$ → $T(n) = O(\log n)$ : 이진탐색
- $T(n) = 2T(n/2)+O(n)$ → $T(n) = O(n\log n)$ : 퀵, 합병

☐ 안정적/제자리 정렬 알고리즘의 개념과 종류

- 안정적 : 동일한 키를 갖는 레코드 쌍의 상대적인 위치가 정렬 후에도 그대로 유지되는 방식
  (안정적 아닌 것은 기준 키 값을 비교 할 때 멀리 떨어진 원소와 비교함, 선택, 셀, 힙, 퀵)
- 제자리 : 입력배열 말고 별도의 원소를 저장하기 위하여 추가적으로 메모리를 필요로 하는 방식
  (비교기반의 합병, 분포기반의 계수, 버킷, 기수)

☐ 비교기반/분포기반 알고리즘의 종류

- 비교기반 : 선택, 버블, 삽입, 셀, 합병, 퀵, 히프
- 분포기반 : 계수, 버킷, 기수

☐ 선택정렬 및 버블정렬의 수행(처리) 방식

- 기준 키값을 미정렬 데이터 중, 맨 앞에 것으로 정한 후 미 정렬 데이터중에서 더 작은 것이 있다면 mim=j를 기억 후, A[I]와 A[min]을 교환하기 때문에 멀리 떨어진거와 교환 되는것이 특징 임. 모든 항목은 기껏해야 2번 이동 함, 실행시간이 입력자료에 민감하지 않다.
- 버블 : 앞에서부터 이웃한 원소와 비교 하게되면 맨 뒤에 정렬 된 것이 예시로 보여 짐, 만약 앞쪽에 정렬된 예시로 보여지면 뒤쪽부터 비교하는 방식 임 , 둘 다 제자리 정렬이며, $O(n^2)$임

☐ 삽입정렬 특징 및 성능(데이터의 입력 상태)

- 카드를 하나 뽑아서 바른 위치에 삽입하는 방식으로 제자리를 찾아 갈때까지 원소의 이동이 빈번하여 셀 정렬을 사용 함, 데이터의 입력이 거의 정렬되어 있는 경우는 빠른 수행을 하면서, 완전히 정렬된 데이터가 들어오면 가장 빠른 성능인 O(n) 됨, 평균적인 경우와 최악의 경우 $O(n^2)$이며, 제자리, 안정적 정렬이다.

☐ 셀 정렬의 개념

- 도날드 셀이 고안 함, 삽입정렬의 보완형태로 처음에는 멀리 떨어진 원소와 비교하여 교환, 점차 가까운 원소와 비교하여 교환, 간격 D의 크기(1, 4, 13, 40, 121, 364....예 39개의 데이터가 들어 올 때 왼쪽 순열에서 39보다 적은 13개의 부분으로 나누어 처리 함)에 따라 다양한 성능을 보임, 멀리 떨어진 원소와 비교함으로 안정적이지 못하며, 제자리 정렬, $O(n^2)$이다.

☐ 퀵정렬의 성능과 점화식

- 성능 : 한 번의 분할과 두 번의 재귀호출로 이루어짐, 재귀 호출시 배열 크기가 절반으로 줄어든다면 재귀 호출의 깊이는 logn이 되고, 배열 크기가 1씩 줄어든다면 재귀호출의 깊이는 n번이 된다. 따라서 최선의 경우와 평균은 O(nlogn)이고, 미리 정렬되어 있는(피봇이 항상 부분 배열 최소일 때) 데이터의 경우 최악의 경우인데 $O(n^2)$이 된다. 불 안정적이며, 제자리 정렬이다
- 점화식
  - ▶ 최악의 경우 $T(n) = T(n-1)+O(n) \rightarrow T(n) = O(n^2)$
  - ▶ 평균적/최선적 $T(n) = 2T(n/2)+O(n) \rightarrow T(n) = O(nlogn)$

☐ 히프정렬의 개념

- 완전이진트리로 구축, 최대값 삭제 및 새로운 원소 삽입 용이함, 1차원 배열로 구성 됨, 주어진 1차원 배열을 히프로 만든 후 아래에서 위로, 왼쪽에서 오른쪽으로 구축하는 방법과 입력배열의 원소를 삽입하는 과정을 반복하면서 히프로 구축하는 방법이 있음, 히프가 구축 되면 최대값 제거 후 다시 히프로 구축하면 됨, 위 과정을 반복하여 정렬을 함. 제자리 정렬, 안정적이지 않음, O(nlogn) 임.

☐ 선형시간 알고리즘 종류와 특징, 처리과정

- 계수정렬 : 입력 키 값의 범위를 알고 이을 때 유용 함.
- 버킷정렬 : 주어진 키 값의 범위를 균등하게 나누어 버킷을 만듬, 입력데이터의 범위가 크지 않을 때 유용 함
- 기수정렬 : 자리수로 나누어 낮은 자리부터 높은 자리로 반복하여 정렬, 입력 키값의 자리수가 상수일 때 유용 함. 삽입정렬 방식 이용 함, d자리수 n개의 데이터 정렬하는 경우 O(dn) 이다.
- 모두 다 O(n), 제자리 정렬 아님, 안정적임

☐ 정렬의 특징들

- 최악 시간 복잡도인 알고리즘 종류 기재하기
  - ▶ O(n) : 계수, 버킷, 기수
  - ▶ O(nlogn) : 합병, 퀵(최선의 경우 임), 히프
  - ▶ $O(n^2)$ : 선택, 버블, 삽입, 셀, 퀵(최악의 경우 임)
- 비교기반 : 선택, 버블, 삽입, 셀, 합병, 퀵, 히프
- 분포기반 : 계수, 버킷, 기수
- 안정적 정렬 : 버블, 삽입, 합병, 계수, 버킷, 기수
- 안정적이지 못한 정렬 : 선택, 셀, 퀵, 히프
- 제자리 정렬 : 선택, 버블, 삽입, 셀, 퀵, 히프
- 제자리 정렬 아닌 것 : 합병, 계수, 버킷, 기수

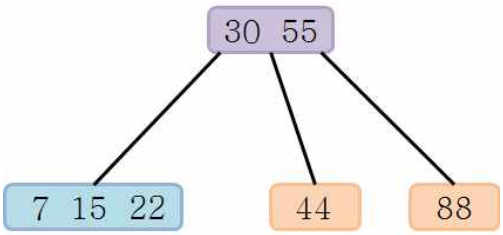
- 입력데이터가 역순으로 정렬 된 경우 최악 : 버블, 삽입, 퀵
- 부분배열로 나누는 것 : 셸, 합병, 퀵
- 내부적으로 다른 정렬을 하는 것 : 셸, 버킷, 기수

☐ 탐색기법의 시간 복잡도
- 순차탐색 : 비 정렬 데이터 탐색에 적합 → O(n)
- 이진탐색 : 정렬되어 있어야 함, 절반씩 나누어 탐색, 삽입/삭제 시 정렬 상태 유지하기위해 자료 이동 필요, O(logn)→ T(n)=T(n/2)+ O(1)
- 이진탐색트리 : 각 노드의 왼쪽 트리는 작고, 오른쪽은 큼, 평균 시간 O(logn), 최악(경사진 경우) → O(n)

☐ 이진탐색 트리의 최악의 경우
- 미리 정렬되어진 데이터 입력 시, 한 쪽으로 경사진 트리나 지그재그 인 트리이며, → 최악의 경우 O(n)이다. 요놈을 해결하기 위하여 균형트리인 2-3-4트리가 나오고 복잡한 3-4노드를 해결하기 위하여 흑적트리가 등장 한다.

☐ 이진탐색트리에서 successor 노드
- 석세스 노드란 후속자, 계승자를 의미하며, 숫자가 뒤 섞여 있더라도 임의의 노드를 선택한 후 석세서 노드가 무엇이냐 문제 나오면 그 숫자 다음 것이 된다. 예) 32, 46, 42, 40, 41 중에 32의 석세스 노드는 40이 된다.

☐ 2-3-4트리의 노두 구성 및 4-노드의 분할
- 아래 그림에서 20이란 값을 삽입 시 루트에서 30보다 작기 때문에 왼쪽을 따라 탐색하기 된다.
- 7, 15, 22 짜리 4노드를 만나게 되면 가운데 2노드로 분할하고 가운데 값 15를 루트 노드로 보냄
- 22 왼쪽에 20을 삽입하여 3노드로 만든다.




☐ 흑적트리의 개념과 성질 회전연산
▶ 개념 : 복잡한 2-3-4노드를 표현하기 어려워 흑적트리 등장 함
▶ 성질
- 모든 노드는 흑색이거나 적색이다
- 루트 노드는 흑색이다
- 자식이 없는 경우 null 노드를 자식으로 가지며 흑색이다
- 적색노드의 두 자식은 항상 흑색이다.
- 임의의 노드부터 리프노드까지 경로 상에는 동일한 흑색노드 개수가 존재 한다.

▶ 연산
- 탐색이 실패한 지점에 적색 노드로 추가 함.
- 적색노드가 연달아 일어나면 흑적트리를 만족시키기 위하여 루트 노드 쪽으로 올라가면서 노드의

색깔을 변경하거나 회전 연산을 한다.

- <참고> 2-3-4트리를 흑적으로 바꾸게 되면 적색노드가 나타나는데, 3노드는 1개가 나타나고, 4노드는 2개가 나타남으로 위 그림에서 적색노드 개수는 3노드 1개, 4노드 1개가 구성되어 있음으로 총 3개의 적색 노드가 생성된다.

☐ 그래프 순회방법의 종류와 개념

- DFS : 최근 정점을 주변으로 우선 방문, 스택사용, 시간복잡도 O(|V|+|E|)
- BFS : 주변 정점 중에서 오래된 것을 먼저 우선 방문, 큐 사용, 시간복잡도 DFS 같음

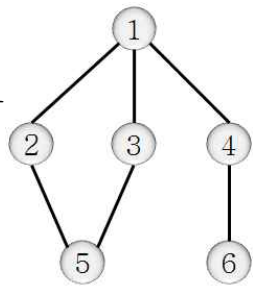
▶ 순회의 응용

- 위상정렬 : 사이클이 없고 간선이 한 방향으로만 된 방향그래프이며 DFS에서 사용
- 연결성분 : 무 방향 그래프에서 임의의 두 정점 간에 경로가 존재하는 최대 부분그래프 DFS/BFS사용
- 강연결성분 : 방향그래프에서 임의의 두 정점을 연결하는 양 방향의 경로가 존재하는 최대의 부분 그래프, DFS사용

- DFS 방문순서

  1 2 5 3 4 6 또는 1 4 6 2 5 3 또는 1 3 5 2 4 5

  <주의> 대부분 왼쪽부터 탐색하는데 시험문제는 최근 정점을 기준으로 하기 때문 반드시 최근 정점을 알아야 함.




- BFS 방문순서

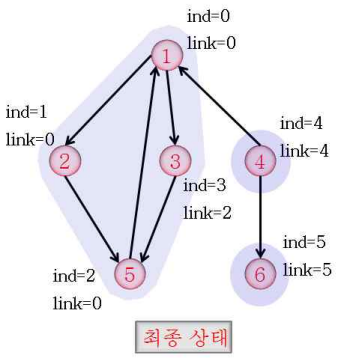
  1 2 3 4 5 6, 또는 1 3 4 2 5 6 또는 1 4 3 2 6 5

  <주의> 선택한지 오래 된 것부터 선택함.

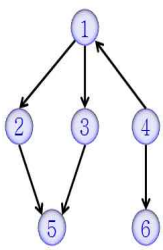
☐ 그래프 순회의 응용(위상정렬, 연결성분, 강 연결성분)

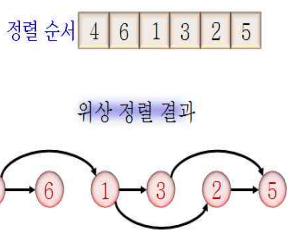
- 위상정렬 : 사이클이 없고 간선이 한 방향으로만 된 방향그래프이며 DFS에서 사용
- 연결성분 : 무 방향 그래프에서 임의의 두 정점 간에 경로가 존재하는 최대 부분그래프 DFS/BFS사용
- 강연결성분 : 방향그래프에서 임의의 두 정점을 연결하는 양 방향의 경로가 존재하는 최대의 부분 그래프, DFS사용
- 강연결성분 개수 예제 : 정답 3개임

<강 연결 성분 예시>




<위상정렬 예시>



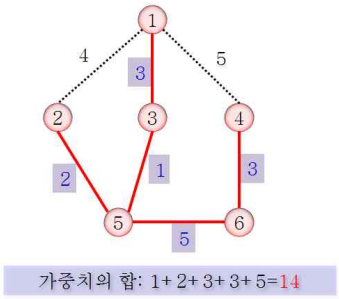
<연결성분 예시>


☐ 최소신장 트리 알고리즘과 구하기

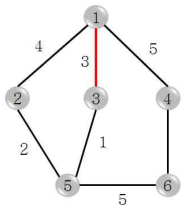
▶ 최소비용 알고리즘

- 사이클이 있으면 안됨
- 크루스칼 : 선택정점은 신경 쓰지말고 가중치가 작은거 부터 구함, O(|E| log |E|)간선의 개수임
- 프림 : 이미 선택된 정점에 속해있는 선택하지 않은 정점 V-S를 잇는 최소의 간선을 선택해 나감
- 둘다 욕심쟁이 기법이며 결과는 같은 값이 나온다.

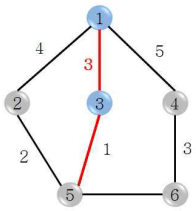
<크루스칼>



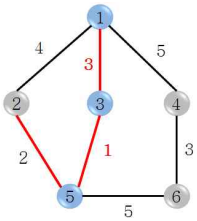

<프림1>




<프림2>




<프림3>




☐ 최단경로 알고리즘의 종류

▶ 데이크스트라(욕심쟁이)

- 특정 하나의 시작 정점에서 다른 모든 정점으로 경로를 구함
- 음의 가중치 없다고 가정, 있으면 제대로 값이 안 나옴
- 거리가 짧은 u를 선택하고 u와 경유하는 기리와 기존 거리 중 작은값을 새 거리 값으로 조정

▶ 플로이드(동적)

- 모든 쌍 간의 최단경로를 구함
- 음수의 사이클이 존재하지 않는다고 가정
- 1인경로부터 시작해서 V까지인 경로까지 단계적으로 해를 구해 나감.

☐ 스트링 매칭 알고리즘 종류와 기본 개념

▶ 종류

- 부루트-포스 : 직선적 검사, 텍스트의 모든 위치에서 시작되는 부분 스트링이 패턴과 일치하는지 검사, O(nm)
  예) 패턴 0001일 때 0000000001 제일 많이 비교, 111111111 제일 적게 비교로 매칭 됨
- 라빈카프 : 스트링을 해시 값으로 계산하여 매치 후보를 찾고, 후보에 대해서만 문자열로 비교해서 매치를 찾는방법, O(nm), 최선의 경우 O(n+m)

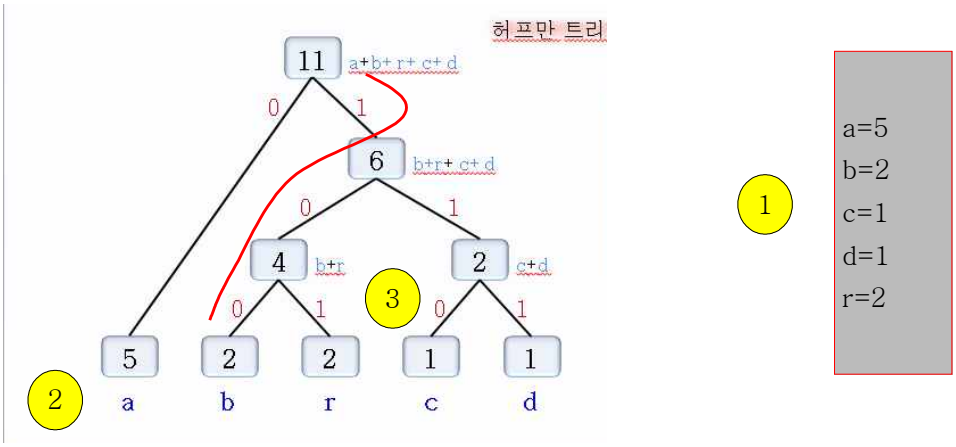
- KMP : 패턴을 전 처리하여 접미부와 일치하는 최대 접두부를 구하고, 이 정보를 이용하여 매칭 함
- 보이어-무어 : 우측에서 좌측으로 진행 함, 중복된 문자를 줄이기 위하여 불일치 문자와 일치 접미부 방법을 사용 함

## 스트링 매칭 알고리즘

| 알고리즘 | 실행 시간 | 특 징 |
|---|---|---|
| 브루트-포스 알고리즘 | O(m(n-m+ 1)) → O(mn) | 텍스트와 패턴이 비슷할수록 비교 회수가 증가<br>최악: T의 매 위치마다 P의 모든 문자를 비교 |
| 라빈-카프 알고리즘 | O(m(n-m+ 1))<br>기대치 O(m+n) | 스트링을 해시값으로 계산하여 매칭<br>modulo 연산 사용 |
| KMP 알고리즘 | O(m+n) | 패턴 전처리 → 최대 접두부 이용 |
| 보이어-무어 알고리즘 | 최악: O(m(n-m+1))<br>최선: O(m+n/m) | 브루트-포스 알고리즘과 유사(단, 문자 비교 방향 : 우측에서 좌측)<br>일치 접미부 방법, 불일치 문자 방법 |

☐ 허프만 코드 구하기

- 스트링에서 각 문자가 나타나는 빈도에 따라 다른 길이를 부호를 부여하는 통계적 압축방법
- 빈도가 높은 문자 → 짧은 문자로 표현, 빈도가 낮은 문자 → 긴 코드로 사용




<설명>

1. 'abracadabra' 문자가 주어지면 노랑색 1번처럼 표를 만든다.
2. 노랑색 2번처럼 알파벳을 나열한다.(주의 : 빈도수가 높은 순으로 나열 함)
3. 노랑색 3번처럼 좌측은 0, 우측은 1로 표시 하며 위쪽으로 올라간다.
4. 예를 들어 b를 구해보자, 최상위 루트부터 빨강색 선을 따라 내려오면 b=100이 된다(나머지 동일)
5. 결과 : a=0, b=100, c=110, d=111, r=101 완성 됨

☐ LZ77 알고리즘의 압축 과정

- 슬라이딩 윈도우로 진행되며 탐색버퍼와 전향버퍼로 구성 됨
- 토큰구성 : (차감거리<오프셋>, 일치거리, 다음입력문자)
- 디코딩 : 탐색 부분에서 일치된 부분과 토큰에 기록된 다음 입력문자를 출력하고, 출력된 문자열을 탐색버퍼 마지막에 추가 함
- 특징: 탐색버퍼는 최근의 문자열로 구성하기 때문에 가까운 부분에서 문자가 반복 될 때에는 좋은 성능을 보임, 일치하는 문자가 없을 경우 압축률이 저하 됨

예 > cabrabadabrarrarrad..... 결과 (3, 2, a)

전향버퍼(슬라이딩 윈도의 오른쪽 부분, abadab)의 처음 두 문자(ab)가 탐색버퍼(슬라이딩 윈도의 왼쪽 부분, cabr)에서 c와 r 사이의 문자와 일치 하는데, 이때 전행 버퍼의 맨 앞으로부터 탐색 버퍼의 일치되는 부분까지의 거리 차이는 3이다, 또한 전향 버퍼에서 2개의 문자가 일치하고, 일치된 두 문자 다음에는 a가 나타난다, 이런 정보를 종합해서 토큰을 구성하면, (3, 2, a)가 된다.

☐ JPEG 표준의 단계별 처리과정


- 순서 암기 : 블록화→DCT→양자화→엔트로피 코딩→허프만 이용 이미지 생성
- 블록화 : 2차원 테이블을 8x8 개로 블럭으로 분할
- DCT : 블록 값을 픽셀을 이용하여 변환
- 양자화 : 큰 값을 작은 값으로 표현(손실이 발생 함)
- 엔트로피 코딩 : 지그재그 순서대로 나열하여 1차원 스트링으로 표현

☐ 동적프로그램 방법의 처리 과정과 해당 알고리즘의 종류

▶ 처리 과정

- 반복된 분할과정을 통해, 얻어진 작은 문제에 대한 결과를 테이블에 저장
- 테이블을 통하여 더 큰 문제의 해를 점진적으로 만들어감.

▶ 알고리즘 종류

- 플로이드 알고리즘 : 모든 쌍 최단경로 구하는문제, $O(n^3)$
- 연쇄행렬 : n개의 행렬곱셈 $M_1$, $M_2$ ...$M_n$에서 최소의 곱셈 횟수를 가진 행렬의 곱셈 순서를 갖는 문제
- 스트링 편집거리 : 두 문자열 X와 Y사이의 편집거리 비용 구하는 알고리즘

☐ NP-완전 문제의 종류와 기본 개념

▶ 기본 개념

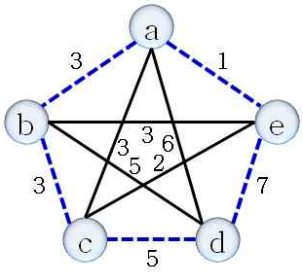
- 클래스 NP에 속하는 모든 문제가 주어진 어떤 문제로, 다항식 시간에 변환되고 그 문제가 클래스 NP에 속하는 경우에 주어진 문제를 NP_완전 문제라 함.
- 아직까지는 다항식 시간 안에 해결하는 알고리즘이 만들어 지지는 않았지만 , 그렇다고 그 알고리즘이 존재하지 않는다고 증명도 안 되었다. 그래서 어려운 문제이다.

▶ NP-완전문제 종류(8개)

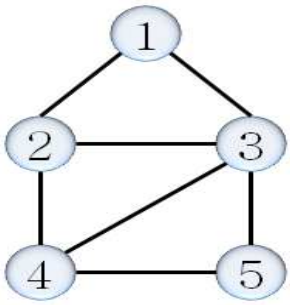
- TSP(외판원 문제), 0/1 배낭 문제, CNF 만족성 문제, 해밀토니언 싸이클 문제, 궤 채우기 문제 파티션 문제, 클리크 판정 문제, 버택스 커버 문제

예)

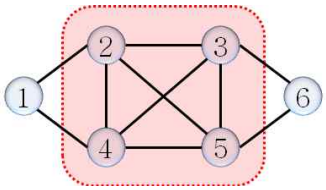
<외판원> 도시를 한번씩 만 지나서 최소비용으로 출발했던 도시로 되돌아오는 문제(a b c d e a)



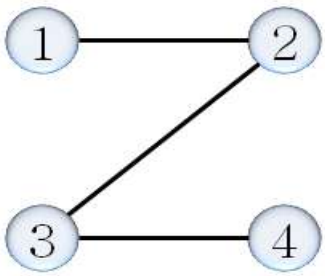

<해밀토니언 사이클> 무방향 그래프에서 모든 정점을 정확히 한 번 만 통과 후 되돌아오는 문제 ( 1 2 4 5 3 1)




<클리크 판정> 빨강색 부분처럼 완전 부분 그래프 문제

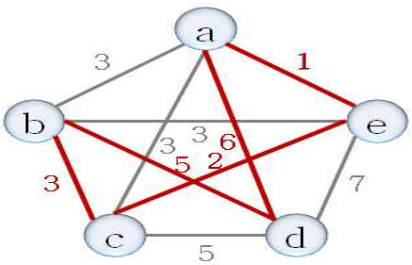



<버택스 커버> 모든 간선이 최소한 하나 이상의 정점에 부수하는 정점의 부분 집합 크기 2 →{2,3}, {1,3}, {2,4}, {1,4}




☐ NP-완전문제 용어

- **클래스 NP** : 비결정론적 튜링기계를 이용하여 다항식 시간에 해결할 수 있는 모든 판정문제의 집합
- 클래스 P : 결정론적 튜링기계를 이용하여 다항식 시간에 해결할 수 있는 모든 판정문제의 집합
- 변환 : B에 대한 알고리즘으로 A문제를 풀 수 있을 때, 문제 A를 문제B로 **변환** 할 수 있다고 하며, 이러한 **변환**에 다항식 시간이 소요되면 **다항식 시간변환**이라 한다.
- NP-완전문제 : NP의 모든 문제가 어떤 문제 A로 다항식 시간내에 변환되고, A가 NP에 속하는 경우
- 최적화문제 : 아래 그림 결과 : 17




<근사알고리즘 최적화 외판원문제> 모든 정점을 한 번씩만 지나가는 싸이클 가중치가 제일 작은 찾음

☐ PRAM 모델의 종류와 특징
- CRCW(Conclusive Read Concurrent Write) : 가장 이상적이고 강력한 방식(C/C)
- CREW(Concurrent Read Exclusive Write) : 가장 현실적인 방식(C/R)
- ERCW(Exclusive Read Concurrent Write) : 현실성이 없는 모델(E/C)
- EREW(Exclusive Read Exclusive Write) : 가장 제한적인 방식(E/E)
- 단어 뜻 Exclusive(독립적인), Concurrent(공존하는, 동시에 발생하는)

☐ 병렬알고리즘의 효율성 평가 척도
- S(n) 정의 : 문제 크기 n에 대한 최선의 순차알고리즘의 수행 시간
- P(n) 정의 : 문제 크기 n에 대한 P개의 프로세서를 사용한 병렬 알고리즘의 수행시간

▶시간 효율성 공식 암기 $$Efficiency = \frac{S(n)/p}{P(n)} = \frac{S(n)}{p \times P(n)}$$

- 시간 효율성은 0-1사이의 값을 가지며, 1인 경우 병렬과정에서 부가적인 시간이 전혀 없다.

▶ 속도 향상률 : 동일한 문제를 푸는 순차 알고리즘에 비하여 몇 배의 속도를 낼 수 있느냐 따진다.
- 속도 향상률 공식 : S(n) / P(n)

▶ 작업량 : 작업량이 S(n)과 같으면 최적의 병렬 알고리즘
- 작업량 공식 : P×P(n)
- 예제) 병렬알고리즘 효율성을 평가하는 척도 중에서, 병렬 알고리즘이 사용한 프로세서의 개수와 그 알고리즘의 시간을 곱한 것을 무엇이라 하는가? **작업량**

☐ 유전알고리즘의 주요 용어/개념
▶ 용어
- 염색체 : 주어진 당면 문제의 탐색 공간에서 하나의 가능한 해법을 함축하고 있음.
- 개체군 : 해법의 집합

▶ 개념 : 염색체 정보의 결합을 통하여 부모보다 더 나은 자손을 생성하기 위해서 해를 선택적으로 키워 나가는 방법(적자생존 원칙에 따름)

☐ 유전알고리즘의 처리 과정
- 처리과정 순서 : **선택 → 교차 → 변이 → 저장**
- 선택 : 적합도에 따라 개체군에서 두 부모 염색체를 선택
  - 룰렛휠 선택 : 염색체의 적합도에 비례하여 개체군에서 다음 세대로 넘겨줄 부모를 선택
  - 순위 선택 : 순위를 정함
  - 토너먼트 선택 : 난수 0, 1발생하여, 기준값보다 작으면 좋은거 선택, 아님 낮은거 선택
  - 엘리트즘 선택: 가장 좋은거 복사, 급격한 성능이 나타남.
- 교차 : 두 부모를 교차시켜 자손을 생성(부모의 형질을 나누어 가짐)
- 변이 : 확률에 따라 새 자손의 염색체의 선택된 위치의 값을 조정(정상보다 빨리 나오지 못하게 억제)
- 저장 : 새롭게 태어난 강한 자식을 새로운 개체군에 포함시킴