Chapter 1. 알고리즘과 문제의 분석 : 원리와 예제 (part 1)

소개

어떤 문제를 **알고리즘적**으로 해결 가능하다 -> 충분한 수행 시간, 기억 공간이 제공될 때 **어떠한 입력에 대해서도 항상 옳은 결과를 산출**하는 컴퓨터 프로그램 작성 가능

이론적으로 수행은 가능하지만 기억 공간이나 수행 속도때문에 현실적이지 않은 것도 있음 -> **계산 복잡도(Computational Complex)** 해야함

수학적 배경

기본적인 수학 내용 복습

집합, 튜플 및 관계

- "$e \in S$ "
- 원소 e는 집합 S의 원소이다." 혹은 "e는 S에 속한다"

  이 경우 e와 S는 서로 다른 자료형 -> e가 정수라면 S는 정수의 집합, 정수와 정수의 집합은 서로 다름

- 집합의 표현

  $S_1 = \{a, b, c\}, S_2 = \{x | 2^n, n \in (1, \ldots, n)\}, S_3 = \{1, \ldots, n\}$
- 어떤 집합 S_1 의 모든 원소들이 다른 집합 S_2 의 원소일 때 -> S_1 을 S_2 의 **부분집합**

  부분집합의 표기 : $S_1 \subseteq S_2$

- 특정한 순서로 된 원소의 모임 : **수열**

  수열의 표기: $(a, b, c), (b, c, a), (c, a, b)$ -> 수열은 순서를 가지므로 각각 서로 다른 수열

- 집합 S 의 원소들이 어떤 정수 n 에 대해서 집합 $\{1, \ldots, n\}$ 과 일대일 대응 : 집합 S 는 **유한하다**

  $|S|$ 를 집합 S 의 원소의 개수 라고 할 때, $|S| = n$

  유한 수열의 모든 원소들이 다르다면, 그 수열을 같은 원소로 구성된 유한 집합에 대한 **순열** 이라고 한다

- n 개의 원소로 된 집합은 $n!$ 개의 서로 다른 순열을 갖는다

- n 개의 원소를 갖는 집합의 부분 집합 개수 :2^n 개

- n 개의 원소를 갖는 유한 집합 S에서 원소의 개수가 k 개인 부분 집합의 개수 : **이항 계수**

  n 개의 항목으로부터 k 개의 항목을 동시에 뽑는 조합의 수 : $\binom{n}{k}$ 혹은 $C(n, k)$

  $C(n, k) \equiv \binom{n}{k} = \frac{n(n-1)\ldots(n-k+1)}{k!}$

  $\sum_{k=0}^{n} \binom{n}{k} = 2^n$

- **튜플** 이란 자료형이 다를 수 있는 원소들의 유한 수열

  두 집합 S 와 T 의 **외적(Cross Product)** 는 튜플의 첫 번째 원소는 집합 S 에서, 두 번째 원소는 T 에서 선택하여 만든 쌍의 집합

  $S \times T = \{(x, y) | x \in S, y \in T\}$

  $|S| \times |T| = |S| \times |T|$

- **관계(relation)** : 어떤 외적의 부분 집합

  가장 중요한 경우는 **이진관계** ($\{(x, y) | x \in R, y \in R, x < y\}$ 같은)

  관계의 주요 성질

  - 모든 $x \in S$ 에 대하여 $(x, x) \in R$ 이면, R 은 반사적(reflexive)이다
  - $(x, y) \in R$ 일 때, 항상 $(y, x) \in R$ 이면 R 은 대칭적(symmetric)이다
  - $(x, y) \in R$ 일 때 항상 $(x, y) \notin R$ 이면, R 은 반대칭적(antisymmetric)이다
  - $(x, y) \in R$ 이고, $(y, z) \in R$ 일 때, $(x, z) \in R$이면 R 은 이행적(transitive)이다(
  - 반사적, 대칭적, 이행적 성질을 만족하는 관계를 **동치 관계** 라고 하며 $\equiv$ 로 표기한다
  - 이진 관계는 원소들이 순서쌍으로 된 집합이므로 각 행이 하나의 튜플을 포함하고 있는, 두 개의 열을 갖는 테이블로 생각
    - 함수는 단순히 테이블의 첫 번째 열에 같은 원소가 두 번 이상 반복적으로 나타나지 않는 관계
    - 이진 관계를 포함하는 많은 문제들은 그래프에 관한 문제로 간주될 수 있음

대수학 및 미적분학 도구

- 내림함수와 올림 함수

  임의의 실수 x에 대해 $\lfloor x \rfloor$ ("x의 내림 정수") 는 x보다 작거나 같은 정수 중에 가장 큰 정수

  임의의 실수 x에 대해 $\lceil x \rceil$ ("x의 올림 정수") 는 x보다 크거나 같은 정수 중에 가장 작은 정수

- 로그

  $b > 1$ 이고 $x > 0$ 일 때, $\log_b x$ 는 $b^L = x$ 를 만족하는 실수 L 이다. 즉, $\log_b x$ 는 b 를 몇 번 제곱해야 x 가 되는가를 나타낸다

로그의 성질

- x, y 를 임의의 양의 실수, a를 임의의 실수, b와 c는 각각 $b > 1, c > 1$을 만족할 때
- $\log_b$ 는 단조증가 함수이다. 즉, $x > y$ 이면 $\log_b x > \log_b y$ 이다.
- $\log_b$ 는 일대일 함수이다. 즉, $\log_b x = \log_b y$ 이면, $x = y$ 이다.
- $\log_b 1 = 0$
- $\log_b b^a = a$
- $\log_b (xy) = \log_b x + \log_b y$
- $\log_b x^a = a \log_b x$
- $x^{\log_b y} = y^{\log_b x}$
- 밑을 변환하기 위한 식 : $\log_c = \log_b x / \log_b c$

- 순열

n개의 서로 다른 객체에 대한 순열은 각 객체를 한 번씩 포함하는 수열

$S = \{s_1, s_2, \ldots, s_n\}$ 이라할 때, S의 순열은 집합 $\{1, 2, \ldots, n\}$으로부터 자기 자신과 대응되는 일대일 함수 π 이다

- π는 i 번 째 원소 s_i를 $\pi(i)$ 번 째 위치로 이동 시켜서 S 를 재배열하는 것으로 생각 가능
- 예를 들어 $n = 5$ 인 경우,$\pi = (4, 3, 1, 5, 2)$는 S의 원소를 $\{s_3, s_5, s_2, s_1, s_4\}$ 로 재배열한다
- n개의 서로 다른 객체에 대한 순열의 수는 $n!$

- 확률

주어진 상황에서 어떤 사건(event)이 k 개의 결과 $s_1, s_2, \ldots, s_k$ 중 오직 하나만 가질 수 있다고 가정
이러한 결과들을 **기본 사건(elementary events)** 라고 함
기본 사건들의 집합 **전체 사건 혹은 전체 집합 (universe)** , U 로 표기
각 결과 s_i 에 대한 확률 $Pr(s_i)$

- $1 \leq i \leq k$ 에 대해 $0 \leq Pr(s_i) \leq 1$
- $Pr(s_1) + Pr(s_2) + \cdots + Pr(s_k) = 1$

$Pr(s_i)$ 는 전체 반복 시행 횟수에 대한 사건 s_i 가 일어나는 횟수의 기대치
$s_1, s_2, \ldots, s_k$ 중에서는 오직 하나만 일어날 수 있으므로 서로 **상호 배제** 의 관계

확률에 대한 여러 정의

- 조건부 확률
  $Pr(S|T) = \frac{Pr(SandT)}{Pr(T)} = \frac{\sum_{s_i \in S \cap T} Pr(s_i)}{\sum_{s_j \in T} Pr(s_j)}$
- 확률적 독립성
  두 사건 S, T가 주어졌을 때, $Pr(SandT) = Pr(S) \times Pr(T)$ 이면, S, T 를 서로 확률적으로 독립이라고 한다
- 기대값과 조건 기대값
  $f(e)$ 를 기본 사건 $e \in U$ 의 집합 상에서 정의된 랜덤 변수라 할 때, f의 기댓값을 $E(f)$라고 표기하며 다음과 같이 정의함
  $E(f) = \sum_{e \in U} f(e)Pr(e)$ (이를 f 의 평균값 이라고 부르기도 함)
  사건 S 에 대한 f 의 조건 기댓값은
  $E(f|S) = \sum_{e \in U} f(e)Pr(e|s) = \sum_{e \in S} f(e)Pr(e|S)$
  - $E(f+g) = E(f) + E(g)$
  - $E(f) = Pr(S)E(f|S) + Pr(\not S)E(f|\not S)$

- 합산과 급수
    - 산술 급수 : 연속된 정수의 합
      $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
    - 다항 급수 : 우선 다음 제곱수의 합을 생각
      $\sum_{i=1}^{n} i^2 = \frac{2n^3+3n^2+n}{6}$
    - 2의 제곱승 : 기하 급수에서 자주 나타남
      $\sum_{i=1}^{n} i^k \approx \frac{1}{k+1} n^{k+1}$
    - 기하 급수
      $\sum_{i=0}^{k} ar^i = a(\frac{r^{k+1}-1}{r-1})$
    - 조화 급수
      *$\sum_{i=1}^{n} \frac{1}{i} \approx \ln(n) + \gamma, \quad \gamma \approx 0.577$
    - 산술 - 기하 급수 : 항 i는 산술 급수, 항 2^i는 기하 급수
      $\sum_{i=1}^{k} i2^i = (k-1)2^{k+1} + 2$
    - 피보나치 수
      $F_n = F_{n-1} + F_{n-2}, \quad n \geq 2$,
      $F_0 = 0, F_1 = 1$

- 단조 증가 함수와 볼록 (Convex) 함수

    - 단조 증가와 단조 감소 함수
      $x \leq y$ 일 때, 항상 $f(x) \leq f(y)$이면 $f(x)$를 **단조 증가 함수** 라고 한다
      $-f(x)$가 단조 증가 함수라면 $f(x)$ 는 단조 감소 함수
      ex) $x \geq 0$ 에서 x, x^2, $x > 0$ 에서 $\log(x), e^x$

    - 선형 보간 함수
      주어진 두 점 u와 v 사이에서 $(u < v)$ 주어진 함수 $f(x)$의 선형 보간 함수
      $L_{f,u,v}(x) = \frac{(v-x)f(u)+(x-u)f(v)}{(v-x)}$
      $= f(u) + (x-u)\frac{f(v)-f(x)}{v-u} = f(v) - (v-x)\frac{f(v)-f(u)}{v-u}$
      즉, $f(u)$와 $f(v)$ 사이를 잇는 선분

    - 볼록 함수
      모든 $u < v$에 대해, 구간 (u, v) 상에서 $f(x) \leq L_{f,u,v}(x)$ 일 때, 함수 $f(x)$를 볼록 함수 라고 한다
      단조 증가이지만 볼록하지 않을 수도 있고, 볼록하지만 단조 증가가 아닐 수도 있음

        - $f(x)$ 가 실수상에서 정의된 연속함수일 때, $f(x)$ 가 볼록 함수이기 위한 필요충분조건
          모든 x, y에 대해, $f(\frac{1}{2}(x+y)) \leq \frac{1}{2}((f(x) + f(y))$
          *x, y*의 중점에서 계산된 f의 값이 *x,* y 사이의 f의 선형 보간 함수의 중점상에 있거나 그 아래에 있다는 것 *(*$f(x)$ 와 $f(y)$ 의 평균*)*

        - 정수 상에서 정의된 함수 $f(n)$이 볼록 함수가 되기 위한 필요충분조건
          모든 $n, n+1, n+2$ 에 대해, $f(n+1) \leq \frac{1}{2}(f(n) + f(n+2))$
          $f(n+1)$ 이 $f(n)$ 과 $f(n+2)$ 의 평균값보다 작거나 같다

        - 단조 증가성과 볼록 함수의 유용한 성질

- $f(n)$ 을 정수상에서만 정의된 함수라 하고, $f^*(x)$를 연속된 정수 사이에서 선형 보간 함수에 의한 f 의 확장 함수라 한다
  a. $f(n)$ 이 단조 증가이면, $f^*(x)$ 가 단조 증가이고, $f^*(x)$ 가 단조 증가이면 $f(n)$ 도 단조 증가이다
  b. $f(n$ 이 볼록하면, $f^*(x)$ 가 볼록이고, $f^*(x)$ 가 볼록이면, $f(n)$ 도 볼록이다
- $f(x)$의 일차 도함수가 존재하면서 음이 아니라면, $f(x)$ 는 단조 증가이다
- $f(x)$의 일차 도함수가 존재하면서 단조 증가라면, $f(x)$는 볼록이다
- $f(x)$의 이차 도함수가 존재하면서 음이 아니라면, $f(x)$는 볼록이다

논리의 기본 원리

논리는 자연어로 된 서술문(문장 혹은 문)을 공식화하기 위한 시스템
정확한 추론을 하기 위해 사용

- 기본 논리 연산자 및 논리
  - $\wedge$ (and), $\vee$ (or), $\neg$ (not)
  - $\rightarrow$ 혹은 $\Rightarrow$: (의미한다 혹은 ~이면 ~이다)
  - $\equiv$: 좌 우식이 논리적으로 동일하다
  - 드모르간의 법칙
    $\neg(A \wedge B) \equiv \neg A \vee \neg B$,
    $\neg(A \vee B) \equiv \neg A \wedge \neg B$

- 한정 기호
  - $\forall$: 일반(*universal*) 한정기호 (모든 x에 대하여)
  - $\exists x$: 존재(*existential*) 한정기호 (x가 존재한다)

- 한정문의 부정과 반례
  $\forall x(A(x)) \Rightarrow B(x))$이 거짓임을 증명
    $\neg(\forall x(A(x) \Rightarrow B(x))) \equiv \exists x \neg(A(x) \Rightarrow B(x))$
    $\equiv \exists x \neg(A(x) \vee B(x))$
    $\equiv \exists x(A(x) \wedge \neg B(x))$
    $A(x)$ 가 참이고 $B(x)$ 가 거짓인 어떤 원소 x 가 존재한다는 것을 보이기만 하면 됨
    이러한 원소 x 를 **반례(counterexample)** 라고 함

- 대우(contrapositive)
  $A \Rightarrow B \equiv (\neg B) \Rightarrow (\neg)A$
  논리식을 대우에 의해 증명하는 것을 모순에 의한 증명*(Proof by contradiction)* 혹은 대우에 의한 증명*(Proof by contraposition)* 이라 한다
- 모순에 의한 증명
  $A \Rightarrow B$ 를 증명하기 위해 부가적인 가정 $\neg B$ 를 첨가하여 B 자체를 증명
    $A \Rightarrow B \equiv (A \wedge \neg B) \Rightarrow B$