時間複雜度與枚舉

關於這堂課

- 學習重點
  - 時間複雜度
    - Big-O Notation
  - 暴力枚舉
    - 迴圈
      - 所有可能, 所有區間
    - 遞迴
      - 全排列, 全組合
  - 排序
  - 二分搜
  - 雙指標

Big-O Notation

找到最接近的兩個數字差

- 子題配分
  - 50%: 2 ≤ n ≤ 100
  - 30%: 2 ≤ n ≤ 10000
  - 20%: 2 ≤ n ≤ 1000000
- 時間限制 1 秒

```
int f(int a[], int n) {
  int ans = INT_MAX;
  for (int i=0; i<n; i++){
    for (int j=0; j<i; j++){
      int tmp = abs(a[i] - a[j]);
      ans = min(ans, tmp);
    }
  }
  return ans;
}
```

Q: 這樣的程式可以得到幾分？

找到最接近的兩個數字差

- T(n) : 執行 f(n) 所用到的時間
  - T(n) = _____

- 當 n 很大，找出最接近的數字

A. n^3

B. n^2

C. n

```
int f(int a[], int n) {
  int ans = INT_MAX;
  for (int i=0; i<n; i++){
    for (int j=0; j<i; j++){
      int tmp = abs(a[i] - a[j]);
      ans = min(ans, tmp);
    }
  }
  return ans;
}
```

Big-O Notation: 用來化簡分析

$T(n) = 3n^2 + 15n + 10$

移除低次方項 / 估計上限

$T(n) \sim= 4n^2$

移除首項係數

$T(n) \sim= n^2$

$T(n) \in O(n^2)$

唸作：T(n) 屬於 big-O of n^2

Practice: 以 big-O 表示下列的式子

- $n^3 + 100n + 1$
- $n * (n-1) / 2 + 10n$
- $(n + 3) * (n^2 + 5)$
- $2^n + n^{10}$
- $n^{1.5} + n + 128$
- $n \log n + n * n^{0.5}$
- $\log_2 n + \log_3 n$

級數的 big-O：先估計上限

- $1 + 2 + 3 + ... + n < n^2$
- $1^2 + 2^2 + 3^2 + \ldots n^2 < n^3$
- $1 + 2 + 4 + \ldots + 2^n < 2^{n+1}$
- $n + n/2 + n/4 + n/8 + \ldots + 1 < 2n$
- $1 + 1/2 + 1/3 + 1/4 + \ldots + 1/n < \lg n + 10$

Time / Space Complexity

- Time Complexity (時間複雜度)
  - 數入量為 n 時，程式執行指令數量的 big-O 表示法
- Space Complexity (空間複雜度)
  - 數入量為 n 時，程式使用空間的 big-O 表示法

Example：Bubble Sort

- BubbleSort 的時間複雜度？
  A. $O(n^2)$
  B. O(n log n)
  C. O(n)
- BubbleSort 的空間複雜度？
  A. $O(n^2)$
  B. O(n)
  C. O(1)

```
void BubbleSort(int a[], int
n){
  for (int i=0; i<n-1; i++){
    for (int j=0; j<n-1-i;
j++){
      if (a[i] < a[i+1]){
        swap(a[i], a[i+1]);
      }
    }
  }
}
```

Example：找出二進位最高位

- 函式 f 的時間複雜度？
  A. O(n)
  B. O(lg n)
  C. O(1)

```
int f(int n) {
  int cnt = 0;
  while (n > 0) {
    n = n / 2;
    cnt++;
  }
  return cnt;
}
```

Example：求級數

- 函式 f 的時間複雜度？
  - $O(n^2)$
  - $O(n)$
  - $O(1)$

```
int f(int n) {
  int sum = 0;

  for (int i=0; i<n; i++){
    sum = sum + i * i;
  }

  return sum;
}
```

Example : 總和最大的前綴

- 函式 f 的時間複雜度？
  - max_element 的時間複雜度？

```
int f(int a[], int n){
  for (int i=0; i<n; i++){
    a[i] += a[i-1];
  }
  return *max_element(a, a+n)
}
```

Example：找出最小的重複數字

- 函式 f 的時間複雜度？
  - C++ sort 的時間複雜度?

```
int f(int a[], int n, int k){
  sort(a, a+n);
  for (int i=0; i<n; i++){
    if (a[i] == a[i-1]) {
      return a[i];
    }
  }
  return -1;
}
```

109 台大資工碩班入學考試 (多選題)

1. (3 points) Let $f(n)$ be the number of additions in the following algorithm.

```
for i = 1 to n do
  for j = 1 to i do
    for k = 1 to j do
      C[i][j][k] = A[i][j][k] + B[i][j][k];
    end for
  end for
end for
```

1. $f(n) = O(n^2\sqrt{n})$
2. $f(n) = O(n^2 \log n)$
3. $f(n) = O(n^3)$
4. $f(n) = O(n^3 \log n)$
5. $f(n) = O(n^{100})$

暴力枚舉

ZeroJudge a147: Print it all

- 列出所有小於 n 且不被 7 整除的正整數。
- 1 ≤ n ≤ 10000

輸入:
10

輸出:
1 2 3 4 5 6 8 9

ZeroJudge b557: 直角三角形

- n 個相異長度木棒, 第 i 個木棒長度為 ai。問組成直角三角形方法數。
- $1 \le n \le 100, 1 \le a_i \le 100$

輸入:
6
3 3 4 4 5 5

輸出:
8

ZeroJudge a059: 完全平方和

- 計算 a 到 b 之間平方數的和
- 1 ≤ a ≤ b ≤ 1000

輸入:
5 35

輸出:
50

ZeroJudge b001: K-間隔子字串

- 給定正整數 K 和字串 S，找出在此字串中有幾個 K-間隔子字串。
- K-間隔子字串為 UVU, 其中 |U| > 0 且 |V| = K。
- 1 ≤ K ≤ 10, 3 ≤ |S| ≤ 1000

bbaabaaaaa bbaabaaaaa
bbaabaaaaa bbaabaaaaa
bbaabaaaaa bbaabaaaaa
bbaabaaaaa

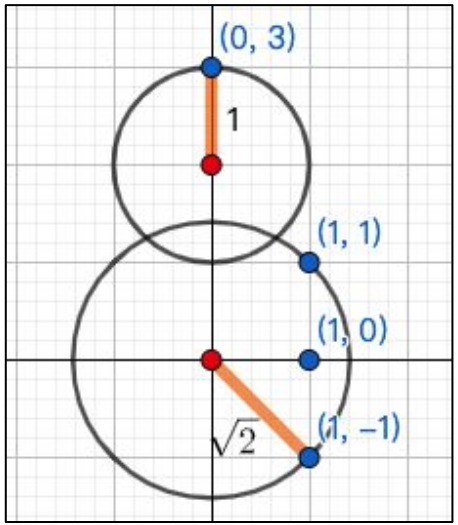
輸入:
1 bbaabaaaaa

輸出:
7

只列舉 d-1 維度

ZeroJudge a480: 導彈攔截系統

- 給兩圓 c1, c2 圓心和 n 個二維座標點 (xi, yi), r1, r2 為 c1, c2 的半徑, 求 r1 * r1 + r2 * r2 最小使得所有點都被其中一個圓蓋住。
- $1 \le n \le 10^6$, |座標點| $\le 10^4$



輸入:
0 0
0 2
4
0 3
1 -1
1 0
1 1

輸出:
3

題單

- 列舉整個範圍
  - zj a147: Print it all
  - zj a040: 阿姆斯壯數
  - zj d299: 程式設計師的面試問題
  - zj b511: 換銅板
  - zj c317: 硬幣問題！前傳
- 迴圈
  - zj b557: 直角三角形
  - zj a583: 1. 座位距離計算問題
  - zj d809: 黑暗土地
  - zj c316: 最遠點對！前傳
- 減少列舉
  - zj a059: 完全平方和
  - zj c777: 106北二2.規律的數列
  - zj a215: 明明愛數數
- 列舉區間
  - zj b001: K-間隔 (K-GAP) 子字串
- 只列舉 d-1 維度
  - zj a480: 導彈攔截系統
- 其他
  - zj b537: 分數運算-1

暴力枚舉 - 使用遞迴

ZeroJudge d471: 0 與 1 的遊戲

- 印出 n 個 bit 所能表示的二進位數字。
- 1 ≤ n ≤ 15

輸入:	輸出:
2	00 01 10 11

ZeroJudge d471: 0 與 1 的遊戲

- 方法一:使用迴圈搭配二進位

```
void solve(int n) {
    int N = (1 << n);  // 2**n
    for (int S = 0; S < N; S++) {
        for (int i = n - 1; i >= 0; i--) {
            if (S & (1 << i)) {
                cout << '1';
            } else {
                cout << '0';
            }
        }
        cout << '\n';
    }
}
```

ZeroJudge d471: 0 與 1 的遊戲

- 方法二:使用遞迴

```
int n;
string s;

void print_all(int top) {
    if (top == n) {
        cout << s << '\n';
    } else {
        s[top] = '0';
        print_all(top + 1);
        s[top] = '1';
        print_all(top + 1);
    }
}
```

```
int main() {
    while (cin >> n) {
        s.resize(n);
        print_all(0);
    }
    return 0;
}
```

ZeroJudge d115: 數字包牌

- 由小到大印出 n 個數字取 m 個的所有組合。
- $1 \le m \le n \le 100$

輸入:
5 5 2 3 9 12 2

輸出:
2 3
2 5
2 9
2 12
3 5
3 9
3 12
5 9
5 12
9 12

ZeroJudge d115: 數字包牌

```
int n, m;
int a[100];
int ans[100];
bool used[100];

void print_all(int last, int top) {
    if (top + n-last < m) return;
    if (m == top) {
        for (int i = 0; i < m; i++)
            cout << ans[i] << ' ';
        cout << '\n';
    } else {
        for (int i = last + 1; i < n; i++) {
            if (used[i] == true) continue;
            used[i] = true; ans[top] = a[i];
            print_all(i, top + 1);
            used[i] = false; ans[top] = 0;
        }
    }
}
```

```
int main() {
    cin.tie(0);
    cin.sync_with_stdio(0);

    while (cin >> n && n != 0) {
        for (int i = 0; i < n; i++) {
            cin >> a[i];
        }
        cin >> m;
        sort(a, a + n);
        print_all(-1, 0);
        cout << endl;
    }
    return 0;
}
```

ZeroJudge d781: Anagram

- 輸入一個字串, 依照字元次序印出所有排列。
  字元次序 : AaBbCcDd.....YyZz

輸入:
acba

輸出:
aabc
aacb
abac
abca
acab
acba
baac
baca
bcaa
caab
caba
cbaa

ZeroJudge d781: Anagram

- 使用 C++ next_permutation

```
bool cmp(char a, char b) {
    if (tolower(a) != tolower(b)) {
        return tolower(a) < tolower(b);
    }
    return a < b;
}

void solve() {
    string str;
    cin >> str;
    sort(str.begin(), str.end(), cmp);
    do {
        cout << str << '\n';
    } while (next_permutation(str.begin(), str.end(), cmp));
}
```

ZeroJudge a229: 括號匹配問題

- 輸出所有 n 個括號的合法匹配字串。
- 1 ≤ n ≤ 13

輸入:	輸出:
3	((()))
	(()())
	(())()
	()(())
	()()()

題單 - 遞迴暴力枚舉

- 列舉子集合
  - zj d471: 0 與 1 的遊戲
- 列舉組合數
  - zj d115: 數字包牌
- 列舉排列
  - zj d781: 00195 - Anagram
  - zj a524: 手機之謎
- 其他
  - zj a229: 括號匹配問題
  - zj a981: 求和問題