PCCA WINTER DAY 1

fsh0524

大綱

- 演算法和複雜度
- 二分搜尋法
- 輸入輸出

演算法和複雜度

什麼是演算法？

什麼是演算法？

- 解決問題的方法與步驟
- 對於同樣一個問題，可能會有很多種解決方案，我們希望選擇效率最高的那個。（較容易，花費時間、空間較少）
- 我們需要分析不同做法的複雜度

如何估計複雜度？

如何估計複雜度？

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

這是經典的泡泡排序，上述的程式最多會進行
(n-1)+(n-2)+(n-3)+...+1 = n*(n-1)/2 次比較和交換
我們可以說這個演算法的複雜度是 O(n^2)

Big-O 函數

$$O(g(n)) = f(n)$$

存在正實數 c, N

使得對於任意正實數 $n \geq N$

$0 \leq f(n) \leq cg(n)$ 成立

Big-O 函數

$$O(2n^2+100n-2000)=O(n^2)$$

$$O(100n)=O(n)$$

$$O(nlgn+n)=O(nlgn)$$

$$O(87)=O(1)$$

二分搜尋法

如何快速地找東西？

全部找一遍：O(n)

如果有單調性呢？

- 假設我們有一個只會回傳 boolean 的 check(x) 函數
- 對資料中的每一個數值 x 都使用 check(x) 函數：
  - False, False, False, …, False, True, True, True, …, True
  - True, True, True, …, True, False, False, False, …, False
- 找符合條件的第一個或最後一個

二分搜尋法

- 輸入：一堆有單調性的資料
- 輸出：符合條件的第一筆或最後一筆資料

- 作法：每次取中間的值，將尋找的範圍縮小一半。

二分搜尋法

```
// 找最後一個符合條件的
int Binary_Search() {
   int L = 0, M, R = INF;
   while (R - L > 1) {
      M = (L + R) / 2;
      if (check(M)) {
         L = M;
      } else {
         R = M;
      }
   }
   return L;
}
```

分析複雜度

每次取中間的值，將尋找的範圍縮小一半

O(log n)

輸入輸出

輸入輸出

- C-style input/output: <cstdio>
- Stream-based input/output: <iostream>
- 常見的題型

C-style input/output

- 格式化輸入：scanf, fscanf, sscanf
- 格式化輸出：printf, fprintf, sprintf
- 非格式化輸入：~~gets~~ (removed in C++14) , fgets
- 非格式化輸出：puts, fputs
- 直接輸入輸出：getchar, putchar

Formatted input/output

%%	一個%
%c	一個字元或一堆字元
%s	一個沒有空白的字串
%[set]	一個只包含 set 中的字元的字串
%d, %i, %u, %o, %x	一個整數
%a, %e, %f, %g	一個浮點數

Stream-based input/output

- std::cin, std::cout
- std::getline, std::istream::getline
- std::fixed, std::scientific
- <iomanip>: std::setw, std::setfill, std::setprecision

常見的題型

- 輸入 T 筆測資 & 輸出到小數點後第 k 位
- 輸入到特定字串為止
- 輸入直到 EOF (End-of-File)
- 輸入整行字串
- 輸入整行未知數量的數列
- 檔案輸入輸出

輸入Ｔ筆測資

輸出到小數點後第ｋ位

cstdio

```
#include <cstdio>

int main() {
    int t;
    double a, b;
    scanf("%d", &t);
    for (int cases = 1 ; cases <= t ;
        ++cases) {
        scanf("%lf%lf", &a, &b);
        printf("Case #%d: %.6lf\n”,
            cases, a+b);
    }
}
```

iostream

```
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    int t;
    double a, b;
    cin >> t;
    for (int cases = 1 ; cases <= t ;
        ++cases) {
        cin >> a >> b;
        cout << "Case #" << cases << ": " <<
            fixed << setprecision(6) <<
            a + b << endl;
    }
}
```

輸入直到 EOF

(End-of-File)

cstdio

```
#include <cstdio>

int main() {
    int a, b;
    while (scanf(“%d%d”, &a, &b) != EOF) {
        printf(“%d\n”, a + b);
    }
}
```

iostream

```cpp
#include <iostream>
using namespace std;
int main() {
    int a, b;
    while (cin >> a >> b) {
        cout << a + b << endl;
    }
}
```

輸入整行字串

fgets

```c
#include <cstdio>
int main() {
    char name[256];
    printf("What is your name? ");
    fgets(name, 256, stdin);
    printf("Hi, %s\n", name);
}
```

std::getline

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string name;
    cout << "What is your name? " << flush;
    getline(cin, name);
    cout << “Hi, ” << name << endl;
}
```

std::istream::getline

```
#include <iostream>
using namespace std;
int main() {
    char name[256];
    cout << “What is your name? ” << flush;
    cin.getline(name, 256);
    cout << “Hi, ” << name << endl;
}
```

輸入以空白分隔的數列

sscanf

```
#include <cstdio>
#include <cstring>
int main() {
    char input[256];
    while (fgets(input, 256, stdin) != NULL) {
        char *ptr = strtok(input," \n");
        int sum = 0, num;
        do {
            sscanf(ptr, “%d”, &num);
            sum += num;
            ptr = strtok(NULL," \n");
        } while (ptr != NULL);
        printf("%d\n”, sum);
    }
}
```

String Stream

```
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
int main() {
    string input;
    stringstream ss;
    int num, sum = 0;
    getline(cin, input);
    ss.clear();
    ss << input;
    while (ss >> num) {
        sum += num;
    }
    cout << sum << endl;
}
```

檔案輸入輸出

freopen

```
#include <cstdio>
int main() {
    int a, b;
    freopen(“test.in”, “r”, stdin);
    freopen(“test.out”, “w”, stdout);
    scanf(“%d%d”, &a, &b);
    printf(“%d\n”, a + b);
    fclose(stdin);
    fclose(stdout);
}
```

fopen

```
#include <cstdio>
int main() {
    int a, b;
    FILE *fin = fopen("test.in", "r");
    FILE *fout = fopen("test.out", "w");
    fscanf(fin, "%d%d", &a, &b);
    fprintf(fout, "%d\n", a + b);
    fclose(fin);
    fclose(fout);
}
```

fstream

```
#include <fstream>
using namespace std;
int main() {
    int a, b;
    fstream fin, fout;
    fin.open(“test.in”, fstream::in);
    fout.open(“test.out”, fstream::out);
    fin >> a >> b;
    fout << a + b << endl;
    fin.close();
    fout.close();
}
```

其他

- std::ios_base::sync_with_stdio
- std::ios::tie
- std::flush, std::endl

參考資料

- 去年的講義
- http://wikipedia.org/
- http://cplusplus.com/
- http://cppreference.com/
- http://chino.taipei/note-2016-0311C-的輸出入cin-cout和scanf-printf誰比較快？/