Contents

1 Shell Script

```
#!/bin/bash
clear
g++ $1.cpp -DDBG -o $1
if [[ "$?" == "0" ]]; then
    echo Running
    ./$1 <$1.in> $1.out
    echo END
fi
```

2 Libraries

2.1 cstdlib

```
#include <cstdlib>
using namespace std;

int main() {
// Function: String conversion
    double atof(const char* str);
        // Convert string to double
    int atoi(const char * str);
        // Convert string to integer
    long int atol(const char * str);
        // Convert string to long integer
    long long int atoll(const char * str);
        // Convert string to long long integer
    double strtod(const char* str, char** endptr);
        // Convert string to double;
    float strtof(const char* str, char** endptr);
        // Convert string to float
    long int strtol(const char* str, char** endptr,
        int base);
        // Convert string to long integer
    long double strtold(const char* str, char**
        endptr);
        // Convert string to long double
    long long int strtoll(const char* str, char**
        endptr, int base);
        // Convert string to long long integer
    unsigned long int strtoul(const char* str, char**
        endptr, int base);
        // Convert string to unsigned long intege
    unsigned long long int strtoull(const char* str,
        char** endptr, int base);
        // Convert string to unsigned long long
            integer
// Function: Integer arithmetics
    int abs(int n);
    long int llabs(long int n);
    long long int llabs(long long int n);
        // Absolute value


    return 0;
}
```

2.2 algorithm

```
#include <algorithm>
using namespace std;

int main() {
    // FI(ForwaradIterator)
    // RAI(RandomAccessIterator)
    // BI(BidirectionalIterator)
    void sort(RAI first, RAI last);

    FI lower_bound(FI first, FI last, const T& k);
    /* 最左邊 ≥ k 的位置 */

    FI upper_bound(FI first, FI last, const T& k);
    /* 最左邊 > k 的位置 */

    pair<FI,FI> equal_range(FI first, FI last, const
        T& k);
    /* 等於 k 的範圍 [lower_bound, upper_bound) */

    bool next_permutation(BI first, BI last);
    /* 使用已經排序(由小到大)的資料，產生下一組排列 */

    bool prev_permutation(BI first, BI last);
    /* 針對逆向排序(由大到小)的資料，產生上一組排序 */

    return 0;
}
```

2.3 map

2.4 set

2.5 vector

2.6 string

3 Algorithms

3.1 最短路

3.1.1 Bellman-Ford

3.1.2 Dijkstra’s

3.2 LIS - Longest Increasing Subsequence

4 Formula

4.1 thm

- 中文測試
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$