TUGAS BESAR

STRATEGI ALGORITMA

ANALISIS ALGORTIMA PENGURUTAN MENGGUNAKAN BRUTE FORCE, GREEDY, BUBBLE SORT, INSERTION SORT DAN MERGE SORT

S1-IF-10-06

Sandy Yopa Boangmanalu	2211102165
Muhammad Fansha Fakhriza	2211102154
Fadlurahman Al Abror	2211102144
Frans Delon Sihite	2211102172
Andarika Syachlimar Restu Rustamaji	2211102173

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

202

DAFTAR ISI

BAB I
DASAR TEORI

1.1. Algoritma Sorting

Algoritma Sorting adalah proses pengurutan data yang sebelumnya disusun secara acak atau tidak teratur sehingga menjadi tersusun secara terurut menurut suatu aturan tertentu atau untuk mengatur elemen-elemen dalam suatu kumpulan data (array atau list) sesuai dengan aturan tertentu, seperti urutan menaik (ascending) atau menurun (descending). Pengurutan sering digunakan dalam berbagai aplikasi, termasuk pengolahan data, pencarian informasi, dan optimalisasi proses.

1.2. Metode Pengurutan Berdasarkan Pendekatan

Metode pengurutan dapat dibedakan berdasarkan pendekatan yang digunakan dalam algoritma. Beberapa pendekatan utama yang digunakan adalah brute force, greedy, dan divide and conquer. Berikut adalah penjelasan masing-masing:

1.2.1 Brute Force

Pendekatan brute force mencoba semua kemungkinan solusi dan memilih solusi yang paling sesuai. Dalam konteks pengurutan, brute force biasanya dilakukan dengan membandingkan setiap elemen dengan elemen lainnya untuk menemukan urutan yang benar. Algoritma ini sederhana namun kurang efisien untuk data berukuran besar.

1.2.2 Greedy

Pendekatan greedy berfokus pada pengambilan keputusan lokal yang optimal pada setiap langkah dengan harapan menghasilkan solusi global yang optimal. Dalam pengurutan, pendekatan ini jarang digunakan secara langsung tetapi dapat diterapkan pada subkomponen algoritma tertentu.

1.2.3 Divide and Conquer

Divide and conquer adalah pendekatan yang membagi masalah menjadi submasalah yang lebih kecil, menyelesaikan masing-masing submasalah, dan menggabungkan hasilnya. Contoh algoritma pengurutan dengan pendekatan ini adalah merge sort.

1.3. Algoritma Brute Force

1.3.1 Bubble Sort

Bubble Sort adalah salah satu metode pengurutan yang sederhana dan paling mudah dipahami, digunakan untuk mengatur elemen-elemen dalam suatu array atau daftar secara berurutan, baik dalam urutan menaik (ascending) maupun menurun (descending). Metode ini bekerja dengan cara membandingkan pasangan elemen yang berdekatan, kemudian menukar posisinya jika elemen yang berada di posisi pertama lebih besar (atau lebih kecil, tergantung urutan yang diinginkan) daripada elemen di posisi berikutnya. Proses ini diulangi secara terus-menerus hingga seluruh elemen dalam daftar berada pada urutan yang diinginkan.

Berikut adalah langkah-langkah dalam algoritma Bubble Sort melalui contoh array "4 2 5 3 9":

Pass Pertama:

- Array awal: (4 2 5 3 9)
- Bandingkan 4 dan 2: karena 4 > 2, tukar posisi → (2 4 5 3 9)
- Bandingkan 4 dan 5: tidak ada perubahan, karena 4 < 5 → (2 4 5 3 9)
- Bandingkan 5 dan 3: karena 5 > 3, tukar posisi → (2 4 3 5 9)
- Bandingkan 5 dan 9: tidak ada perubahan, karena 5 < 9 → (2 4 3 5 9)

Pass Kedua:

- Array awal: (2 4 3 5 9)
- Bandingkan 2 dan 4: tidak ada perubahan, karena 2 < 4 → (2 4 3 5 9)
- Bandingkan 4 dan 3: karena 4 > 3, tukar posisi → (2 3 4 5 9)
- Bandingkan 4 dan 5: tidak ada perubahan, karena 4 < 5 → (2 3 4 5 9)

Pass Ketiga:

- Array awal: (2 3 4 5 9)
- Bandingkan 2 dan 3: tidak ada perubahan, karena 2 < 3 → (2 3 4 5 9)
- Bandingkan 3 dan 4: tidak ada perubahan, karena 3 < 4 → (2 3 4 5 9)
- Setelah pass ketiga, array sudah sepenuhnya terurut.

1.3.2 Insertion Sort

Insertion Sort bekerja dengan membangun hasil urutan secara bertahap. Setiap elemen array dipindahkan ke posisi yang benar relatif terhadap elemen yang sudah terurut. Berikut langkah-langkahnya:

1. Misalkan array "4 2 5 3 9":
2. Elemen pertama (4) dianggap sudah terurut.
3. Elemen kedua (2) dibandingkan dengan 4. Karena 2 < 4, elemen 2 disisipkan sebelum 4: (2 4 5 3 9).
4. Elemen ketiga (5) dibandingkan dengan elemen-elemen sebelumnya. Karena 5 > 4, tidak ada perubahan: (2 4 5 3 9).
5. Elemen keempat (3) dibandingkan dengan elemen-elemen sebelumnya. Disisipkan di antara 2 dan 4: (2 3 4 5 9).
6. Elemen kelima (9) sudah lebih besar dari semua elemen sebelumnya, sehingga tidak ada perubahan.

1.4. Algoritma Greedy

1.4.1 Merge Sort

Merge Sort adalah algoritma berbasis greedy yang menggunakan prinsip divide and conquer. Algoritma ini terdiri dari tiga tahap:

1. Divide (Pembagian): Array dibagi menjadi dua bagian yang hampir sama besar secara rekursif hingga setiap bagian hanya memiliki satu elemen.
2. Conquer (Penguasaan): Subarray yang terdiri dari satu elemen dianggap sudah terurut.
3. Combine (Penggabungan): Subarray-subarray digabungkan kembali dengan cara membandingkan elemen-elemen terdepan dari dua subarray. Elemen yang lebih kecil dipindahkan ke array hasil gabungan.

Misalkan array "4 2 5 3 9":

- Divide: (4 2 5) dan (3 9); kemudian (4 2) dan (5); lalu (3) dan (9).
- Conquer: (4) dan (2) diurutkan menjadi (2 4); (3) dan (9) tetap terurut.
- Combine: (2 4) digabung dengan (5) menjadi (2 4 5); (2 4 5) digabung dengan (3 9) menjadi (2 3 4 5 9)

BAB II
IMPLEMENTASI

2.1 Spesifikasi Hardware dan Software

1. Perangkat Keras *(Hardware)*

   a. Device : Legion 5

   b. Processor : 12th Gen Intel® Core™ i7-12700H 2.30 GHz

   c. RAM : 16,0 GB (15,7 GB usable)

2. Perangkat Lunak *(Software)*

   a. Sistem Operasi : *Windows 11 64-bit*

   b. Bahasa Pemrograman : *C++*

   c. Aplikasi : *Visual studio code*

   d. Compiler/Interpreter *GCC*

2.2 Penjelasan Pseudocode

Pseudocode ini menjelaskan cara kerja dari program yang membandingkan tiga algoritma pengurutan (*Bubble Sort, Quick Sort,* dan *Merge Sort*) berdasarkan waktu eksekusi dengan data acak. Berikut adalah ringkasan *pseudocode* untuk menjelaskan cara kerja program secara keseluruhan:

1. **Impor Library:**

Program dimulai dengan mengimpor pustaka yang diperlukan untuk pembuatan data acak, pengurutan data, pengukuran waktu, dan pembuatan antarmuka pengguna.

```
#include <iostream> # Untuk input dan output data
#include <vector> # Untuk mendeklarasikan dan menggunakan tipe data
#include <string> # Untuk mendukung manipulasi string
#include <random> # Untuk menghasilkan data acak menggunakan generator
#include <chrono> # Untuk mengukur waktu eksekusi algoritma
#include <algorithm> # Untuk fungsi-fungsi manipulasi array dan koleksi
#include <iomanip> # Untuk memformat output, seperti pengaturan presisi angka desimal pada waktu eksekusi
```

2. **Fungsi generate_random_data:**

Fungsi ini menghasilkan sejumlah data acak dalam format string yang terdiri dari satu huruf (karakter acak) dan dua digit angka. Data ini nantinya akan digunakan sebagai input untuk algoritma pengurutan.

```cpp
// Generate random data
std::vector<std::string> generateRandomData(int size) {
    std::vector<std::string> data;
    std::random_device rd;
    std::mt19937 gen(rd());
    std::uniform_int_distribution<> dis(0, 99);

    for (int i = 0; i < size; i++) {
        char letter = 'A' + (rand() % 26);
        std::string item = std::string(1, letter) +
        (dis(gen) < 10 ? "0" : "") +
        std::to_string(dis(gen));
        data.push_back(item);
    }
    return data;
}
```

3. **Define algoritma sorting:**

a. *Bubble Sort*

```cpp
// Bubble Sort
SortResult bubbleSortWithTime(std::vector<std::string> arr) {
    auto start = std::chrono::high_resolution_clock::now();
    std::vector<std::string> data = arr;

    for (size_t i = 0; i < data.size(); i++) {
        for (size_t j = 0; j < data.size() - i - 1; j++) {
            if (data[j] > data[j + 1]) {
                std::swap(data[j], data[j + 1]);
            }
        }
    }

    auto end = std::chrono::high_resolution_clock::now();
    double time = std::chrono::duration<double, std::milli>(end - start).count();
    return {data, time};
}
```

Algoritma *Bubble Sort* bekerja dengan membandingkan pasangan elemen berurutan dalam array. Jika elemen pada indeks ke-i lebih besar dari elemen pada indeks ke-(i+1), maka kedua elemen tersebut akan ditukar posisinya. Proses ini diulang hingga seluruh elemen dalam array terurut.

b. *Insertion Sort*

```cpp
// Insertion Sort
SortResult insertionSortWithTime(std::vector<std::string> arr) {
    auto start = std::chrono::high_resolution_clock::now();
    std::vector<std::string> data = arr;

    for (size_t i = 1; i < data.size(); i++) {
        std::string key = data[i];
        int j = i - 1;
        while (j >= 0 && data[j] > key) {
            data[j + 1] = data[j];
            j--;
        }
        data[j + 1] = key;
    }

    auto end = std::chrono::high_resolution_clock::now();
    double time = std::chrono::duration<double, std::milli>(end - start).count();
    return {data, time};
}
```

Algoritma *Insertion Sort* metode pengurutan yang bekerja seperti cara manusia mengurutkan kartu dalam permainan kartu. Elemen diambil satu per satu dan dimasukkan ke dalam posisi yang sesuai dalam bagian array yang sudah terurut.

c. *Merge Sort*

```
// Merge Sort
SortResult mergeSortWithTime(std::vector<std::string> arr) {
    auto start = std::chrono::high_resolution_clock::now();
    std::vector<std::string> data = arr;

    mergeSortImpl(data, 0, data.size() - 1);

    auto end = std::chrono::high_resolution_clock::now();
    double time = std::chrono::duration<double, std::milli>(end - start).count();
    return {data, time};
}
```

Algoritma *Merge Sort* membagi array menjadi dua bagian lebih kecil, menyortir masing-masing bagian secara rekursif, dan akhirnya menggabungkan kedua bagian tersebut menjadi satu array yang terurut.

4. Fungsi Brute Force Strategy:

Fungsi *bruteForceStrategyWithTime* melakukan pengurutan menggunakan strategi Brute Force dengan langkah berikut:

Mengukur Waktu Eksekusi: Mencatat waktu mulai dan akhir menggunakan *std::chrono* untuk menghitung durasi eksekusi.

Proses Pengurutan:

1. Mencari elemen terkecil dari array menggunakan *std::min_element.*
2. Memindahkan elemen terkecil ke array terurut (sortedData).
3. Menghapus elemen tersebut dari array asli (data).
4. Mengembalikan Hasil: Data yang telah diurutkan dan waktu eksekusi dikembalikan sebagai hasil dalam bentuk struct *SortResult*.

Kelebihan: Sederhana dan mudah dipahami.
Kekurangan: Tidak efisien dengan kompleksitas waktu $O(n^2)$.

```
// Brute Force Strategy
SortResult bruteForceStrategyWithTime(std::vector<std::string> arr) {
    auto start = std::chrono::high_resolution_clock::now();
    std::vector<std::string> data = arr;
    std::vector<std::string> sortedData;

    while (!data.empty()) {
        auto minElement = std::min_element(data.begin(), data.end());
        sortedData.push_back(*minElement);
        data.erase(minElement);
    }

    auto end = std::chrono::high_resolution_clock::now();
    double time = std::chrono::duration<double, std::milli>(end - start).count();
    return {sortedData, time};
}
```

5. Fungsi Greedy:

Fungsi *greedyStrategyWithTime* menerapkan *algoritma selection sort* untuk mengurutkan elemen-elemen dalam sebuah vector berisi string secara leksikografis (ascending order). Fungsi ini juga mengukur waktu eksekusi algoritma menggunakan fitur waktu dari pustaka <*chrono*> di C++.
Berikut adalah penjelasan singkat:

1. Parameter Input:
    - Fungsi menerima sebuah vector string arr sebagai input.

2. Pencatatan Waktu Awal:
    - Waktu awal dieksekusi dengan std::chrono::high_resolution_clock::now().

3. Proses Sorting (Selection Sort):
    - Iterasi melalui seluruh elemen vector.
    - Untuk setiap elemen, fungsi mencari indeks elemen terkecil di antara elemen-elemen yang tersisa.
    - Setelah indeks terkecil ditemukan, elemen tersebut ditukar dengan elemen saat ini.

4. Pencatatan Waktu Akhir:
    - Setelah sorting selesai, waktu akhir dicatat menggunakan *std::chrono::high_resolution_clock::now()*.

5. Penghitungan Waktu Eksekusi:
    - Selisih antara waktu akhir dan waktu awal dihitung dalam milidetik *(std::chrono::duration<double, std::milli>)*.

Hasil Return:
- Fungsi mengembalikan hasil berupa:
- Vector data yang sudah diurutkan.
- Waktu eksekusi algoritma dalam satuan milidetik.

```
// Greedy Strategy (Selection Sort)
SortResult greedyStrategyWithTime(std::vector<std::string> arr) {
    auto start = std::chrono::high_resolution_clock::now();
    std::vector<std::string> data = arr;

    for (size_t i = 0; i < data.size(); i++) {
        size_t minIdx = i;
        for (size_t j = i + 1; j < data.size(); j++) {
            if (data[j] < data[minIdx]) {
                minIdx = j;
            }
        }
        std::swap(data[i], data[minIdx]);
    }

    auto end = std::chrono::high_resolution_clock::now();
    double time = std::chrono::duration<double, std::milli>(end - start).count();
    return {data, time};
}
```

BAB III
PENGUJIAN

3.1 Pengujian Program Sorting Algorithm

1. Data yang digunakan

   Data yang digunakan untuk pengujian adalah data acak berupa string yang terdiri dari satu huruf (karakter acak dari string.ascii_letters) dan dua digit angka (angka acak antara 0 hingga 99). Setiap data yang dihasilkan memiliki format seperti A23, B45, atau Z67. Misalnya, jika ukuran inputnya adalah 100, maka program akan menghasilkan 100 data acak.

2. Pengujian jumlah input yang berbeda

   Pengujian dilakukan dengan ukuran input yang semakin besar, seperti:

   - 100 Iteration
   - 200 Iteration
   - 300 Iteration
   - 400 Iteration
   - 500 Iteration dan seterusnya

   Masing-masing pengujian dilakukan beberapa kali untuk mendapatkan rata-rata waktu eksekusi yang lebih stabil.

3. Pengujian dengan data yang sama untuk semua algoritma

   Untuk memastikan bahwa data yang digunakan untuk pembandingan algoritma adalah data yang sama, dengan menyimpan data acak yang dihasilkan dalam array yang terpisah dan menggunakan data yang sama sebagai input untuk ketiga algoritma. Langkah-langkahnya adalah sebagai berikut:

   - Hasilkan data acak dengan fungsi generate_random_data(size) untuk setiap ukuran input yang diuji.
   - Simpan data acak tersebut dalam array baru.
   - Gunakan data yang sama untuk *Bubble Sort, Insertion Sort, dan Merge Sort.*

4. Catat waktu eksekusi untuk masing-masing Algoritma:

   Setelah data dihasilkan, waktu eksekusi untuk masing-masing algoritma (*Bubble Sort, Insertion Sort, Merge Sort*) dihitung dengan menggunakan fungsi perf_counter(). Program kemudian mencatat waktu rata-rata untuk setiap algoritma berdasarkan jumlah iterasi yang diberikan. Berikut adalah hasil pengujian proses pengukuran data dan waktu eksekusi untuk setiap algoritma:

LAMPIRAN

OUTPUT Proses Problem Size

```
Enter problem size: 10
Enter number of iterations: 100
Original Data:
P27 H042 Q54 G41 H6 U60 M023 E94 A98 Y039


Results for 10 items, averaged over 100 iterations:

Brute Force Strategy:
Average Time: 0.007 ms
Sorted Data:
A98 E94 G41 H042 H6 M023 P27 Q54 U60 Y039

Greedy Strategy:
Average Time: 0.002 ms
Sorted Data:
A98 E94 G41 H042 H6 M023 P27 Q54 U60 Y039

Bubble Strategy:
Average Time: 0.002 ms
Sorted Data:
A98 E94 G41 H042 H6 M023 P27 Q54 U60 Y039

Insertion Strategy:
Average Time: 0.002 ms
Sorted Data:
A98 E94 G41 H042 H6 M023 P27 Q54 U60 Y039

Merge Strategy:
Average Time: 0.010 ms
Sorted Data:
A98 E94 G41 H042 H6 M023 P27 Q54 U60 Y039


Process returned 0 (0x0)   execution time : 8.645 s
Press any key to continue.
```

```
Enter problem size: 20
Enter number of iterations: 200
Original Data:
P31 H22 Q44 G38 H2 U0 M41 E36 A6 Y18 L76 N049 L51 F40 D03 X90 F85 I73 R93 C86


Results for 20 items, averaged over 200 iterations:

Brute Force Strategy:
Average Time: 0.016 ms
Sorted Data:
A6 C86 D03 E36 F40 F85 G38 H2 H22 I73 L51 L76 M41 N049 P31 Q44 R93 U0 X90 Y18

Greedy Strategy:
Average Time: 0.005 ms
Sorted Data:
A6 C86 D03 E36 F40 F85 G38 H2 H22 I73 L51 L76 M41 N049 P31 Q44 R93 U0 X90 Y18

Bubble Strategy:
Average Time: 0.006 ms
Sorted Data:
A6 C86 D03 E36 F40 F85 G38 H2 H22 I73 L51 L76 M41 N049 P31 Q44 R93 U0 X90 Y18

Insertion Strategy:
Average Time: 0.005 ms
Sorted Data:
A6 C86 D03 E36 F40 F85 G38 H2 H22 I73 L51 L76 M41 N049 P31 Q44 R93 U0 X90 Y18

Merge Strategy:
Average Time: 0.020 ms
Sorted Data:
A6 C86 D03 E36 F40 F85 G38 H2 H22 I73 L51 L76 M41 N049 P31 Q44 R93 U0 X90 Y18


Process returned 0 (0x0)   execution time : 9.750 s
Press any key to continue.
```

```
Enter problem size: 30
Enter number of iterations: 300
Original Data:
P82 H13 Q034 G49 H85 U16 M76 E66 A4 Y47 L70 N7 L0 F78 D23 X65 F042 I73 R38 C23 V60 S34
 C53 X96 G22 G15 B31 W74 K2 F97


Results for 30 items, averaged over 300 iterations:

Brute Force Strategy:
Average Time: 0.027 ms
Sorted Data:
A4 B31 C23 C53 D23 E66 F042 F78 F97 G15 G22 G49 H13 H85 I73 K2 L0 L70 M76 N7 P82 Q034
R38 S34 U16 V60 W74 X65 X96 Y47

Greedy Strategy:
Average Time: 0.010 ms
Sorted Data:
A4 B31 C23 C53 D23 E66 F042 F78 F97 G15 G22 G49 H13 H85 I73 K2 L0 L70 M76 N7 P82 Q034
R38 S34 U16 V60 W74 X65 X96 Y47

Bubble Strategy:
Average Time: 0.012 ms
Sorted Data:
A4 B31 C23 C53 D23 E66 F042 F78 F97 G15 G22 G49 H13 H85 I73 K2 L0 L70 M76 N7 P82 Q034
R38 S34 U16 V60 W74 X65 X96 Y47

Insertion Strategy:
Average Time: 0.008 ms
Sorted Data:
A4 B31 C23 C53 D23 E66 F042 F78 F97 G15 G22 G49 H13 H85 I73 K2 L0 L70 M76 N7 P82 Q034
R38 S34 U16 V60 W74 X65 X96 Y47

Merge Strategy:
Average Time: 0.029 ms
Sorted Data:
A4 B31 C23 C53 D23 E66 F042 F78 F97 G15 G22 G49 H13 H85 I73 K2 L0 L70 M76 N7 P82 Q034
R38 S34 U16 V60 W74 X65 X96 Y47


Process returned 0 (0x0)   execution time : 11.294 s
Press any key to continue.
```

```
Enter problem size: 40
Enter number of iterations: 400
Original Data:
P84 H68 Q57 G99 H19 U91 M76 E01 A90 Y34 L21 N97 L31 F36 D68 X5 F91 I020 R49 C11 V19 S9 C059 X3 G13 G13
 B074 W63 K010 F54 N20 Q91 D64 U50 X67 W51 F05 N78 F43 O32


Results for 40 items, averaged over 400 iterations:

Brute Force Strategy:
Average Time: 0.040 ms
Sorted Data:
A90 B074 C059 C11 D64 D68 E01 F05 F36 F43 F54 F91 G13 G13 G99 H19 H68 I020 K010 L21 L31 M76 N20 N78 N9
7 O32 P84 Q57 Q91 R49 S9 U50 U91 V19 W51 W63 X3 X5 X67 Y34

Greedy Strategy:
Average Time: 0.016 ms
Sorted Data:
A90 B074 C059 C11 D64 D68 E01 F05 F36 F43 F54 F91 G13 G13 G99 H19 H68 I020 K010 L21 L31 M76 N20 N78 N9
7 O32 P84 Q57 Q91 R49 S9 U50 U91 V19 W51 W63 X3 X5 X67 Y34

Bubble Strategy:
Average Time: 0.018 ms
Sorted Data:
A90 B074 C059 C11 D64 D68 E01 F05 F36 F43 F54 F91 G13 G13 G99 H19 H68 I020 K010 L21 L31 M76 N20 N78 N9
7 O32 P84 Q57 Q91 R49 S9 U50 U91 V19 W51 W63 X3 X5 X67 Y34

Insertion Strategy:
Average Time: 0.012 ms
Sorted Data:
A90 B074 C059 C11 D64 D68 E01 F05 F36 F43 F54 F91 G13 G13 G99 H19 H68 I020 K010 L21 L31 M76 N20 N78 N9
7 O32 P84 Q57 Q91 R49 S9 U50 U91 V19 W51 W63 X3 X5 X67 Y34

Merge Strategy:
Average Time: 0.035 ms
Sorted Data:
A90 B074 C059 C11 D64 D68 E01 F05 F36 F43 F54 F91 G13 G13 G99 H19 H68 I020 K010 L21 L31 M76 N20 N78 N9
7 O32 P84 Q57 Q91 R49 S9 U50 U91 V19 W51 W63 X3 X5 X67 Y34


Process returned 0 (0x0)   execution time : 10.869 s
Press any key to continue.
```

```
Enter problem size: 50
Enter number of iterations: 500
Original Data:
P49 H036 Q24 G95 H34 U63 M45 E32 A78 Y2 L048 N92 L76 F68 D99 X16 F080 I94 R013 C044 V59 S51 C36 X26 G0
80 G64 B71 W043 K031 F21 N56 Q32 D34 U68 X72 W64 F18 N80 F18 O65 Z63 V18 S3 R32 T89 K52 J6 P43 R10 E98


Results for 50 items, averaged over 500 iterations:

Brute Force Strategy:
Average Time: 0.058 ms
Sorted Data:
A78 B71 C044 C36 D34 D99 E32 E98 F080 F18 F18 F21 F68 G080 G64 G95 H036 H34 I94 J6 K031 K52 L048 L76 M
45 N56 N80 N92 O65 P43 P49 Q24 Q32 R013 R10 R32 S3 S51 T89 U63 U68 V18 V59 W043 W64 X16 X26 X72 Y2 Z63

Greedy Strategy:
Average Time: 0.024 ms
Sorted Data:
A78 B71 C044 C36 D34 D99 E32 E98 F080 F18 F18 F21 F68 G080 G64 G95 H036 H34 I94 J6 K031 K52 L048 L76 M
45 N56 N80 N92 O65 P43 P49 Q24 Q32 R013 R10 R32 S3 S51 T89 U63 U68 V18 V59 W043 W64 X16 X26 X72 Y2 Z63

Bubble Strategy:
Average Time: 0.029 ms
Sorted Data:
A78 B71 C044 C36 D34 D99 E32 E98 F080 F18 F18 F21 F68 G080 G64 G95 H036 H34 I94 J6 K031 K52 L048 L76 M
45 N56 N80 N92 O65 P43 P49 Q24 Q32 R013 R10 R32 S3 S51 T89 U63 U68 V18 V59 W043 W64 X16 X26 X72 Y2 Z63

Insertion Strategy:
Average Time: 0.018 ms
Sorted Data:
A78 B71 C044 C36 D34 D99 E32 E98 F080 F18 F18 F21 F68 G080 G64 G95 H036 H34 I94 J6 K031 K52 L048 L76 M
45 N56 N80 N92 O65 P43 P49 Q24 Q32 R013 R10 R32 S3 S51 T89 U63 U68 V18 V59 W043 W64 X16 X26 X72 Y2 Z63

Merge Strategy:
Average Time: 0.048 ms
Sorted Data:
A78 B71 C044 C36 D34 D99 E32 E98 F080 F18 F18 F21 F68 G080 G64 G95 H036 H34 I94 J6 K031 K52 L048 L76 M
45 N56 N80 N92 O65 P43 P49 Q24 Q32 R013 R10 R32 S3 S51 T89 U63 U68 V18 V59 W043 W64 X16 X26 X72 Y2 Z63


Process returned 0 (0x0)   execution time : 2.904 s
Press any key to continue.
```

```
Enter problem size: 60
Enter number of iterations: 600
Original Data:
P33 H17 Q073 G067 H4 U25 M095 E58 A1 Y21 L83 N81 L90 F01 D46 X68 F87 I83 R28 C24 V49 S19 C10 X39 G59 G
62 B43 W70 K65 F44 N70 Q89 D13 U30 X62 W57 F032 N69 F87 O42 Z69 V0 S7 R039 T61 K051 J76 P23 R80 E88 P1
2 G083 G91 X18 R56 P66 N27 R11 V60 Y35


Results for 60 items, averaged over 600 iterations:

Brute Force Strategy:
Average Time: 0.079 ms
Sorted Data:
A1 B43 C10 C24 D13 D46 E58 E88 F01 F032 F44 F87 F87 G067 G083 G59 G62 G91 H17 H4 I83 J76 K051 K65 L83
L90 M095 N27 N69 N70 N81 O42 P12 P23 P33 P66 Q073 Q89 R039 R11 R28 R56 R80 S19 S7 T61 U25 U30 V0 V49 V
60 W57 W70 X18 X39 X62 X68 Y21 Y35 Z69

Greedy Strategy:
Average Time: 0.033 ms
Sorted Data:
A1 B43 C10 C24 D13 D46 E58 E88 F01 F032 F44 F87 F87 G067 G083 G59 G62 G91 H17 H4 I83 J76 K051 K65 L83
L90 M095 N27 N69 N70 N81 O42 P12 P23 P33 P66 Q073 Q89 R039 R11 R28 R56 R80 S19 S7 T61 U25 U30 V0 V49 V
60 W57 W70 X18 X39 X62 X68 Y21 Y35 Z69

Bubble Strategy:
Average Time: 0.039 ms
Sorted Data:
A1 B43 C10 C24 D13 D46 E58 E88 F01 F032 F44 F87 F87 G067 G083 G59 G62 G91 H17 H4 I83 J76 K051 K65 L83
L90 M095 N27 N69 N70 N81 O42 P12 P23 P33 P66 Q073 Q89 R039 R11 R28 R56 R80 S19 S7 T61 U25 U30 V0 V49 V
60 W57 W70 X18 X39 X62 X68 Y21 Y35 Z69

Insertion Strategy:
Average Time: 0.023 ms
Sorted Data:
A1 B43 C10 C24 D13 D46 E58 E88 F01 F032 F44 F87 F87 G067 G083 G59 G62 G91 H17 H4 I83 J76 K051 K65 L83
L90 M095 N27 N69 N70 N81 O42 P12 P23 P33 P66 Q073 Q89 R039 R11 R28 R56 R80 S19 S7 T61 U25 U30 V0 V49 V
60 W57 W70 X18 X39 X62 X68 Y21 Y35 Z69

Merge Strategy:
Average Time: 0.054 ms
Sorted Data:
A1 B43 C10 C24 D13 D46 E58 E88 F01 F032 F44 F87 F87 G067 G083 G59 G62 G91 H17 H4 I83 J76 K051 K65 L83
L90 M095 N27 N69 N70 N81 O42 P12 P23 P33 P66 Q073 Q89 R039 R11 R28 R56 R80 S19 S7 T61 U25 U30 V0 V49 V
60 W57 W70 X18 X39 X62 X68 Y21 Y35 Z69


Process returned 0 (0x0)   execution time : 11.549 s
Press any key to continue.
```

```
Enter problem size: 70
Enter number of iterations: 700
Original Data:
P30 H35 Q57 G84 H18 U76 M78 E94 A30 Y66 L96 N53 L37 F11 D17 X93 F10 I88 R49 C22 V040 S012 C34 X89 G013
 G15 B48 W48 K51 F08 N10 Q28 D67 U58 X46 W69 F81 N80 F91 O80 Z69 V97 S68 R77 T22 K73 J5 P65 R8 E78 P66
 G77 G74 X38 R41 P31 N58 R26 V34 Y49 S79 T28 M69 W5 C34 Y87 S2 Y88 Y036 C5


Results for 70 items, averaged over 700 iterations:

Brute Force Strategy:
Average Time: 0.105 ms
Sorted Data:
A30 B48 C22 C34 C34 C5 D17 D67 E78 E94 F08 F10 F11 F81 F91 G013 G15 G74 G77 G84 H18 H35 I88 J5 K51 K73
 L37 L96 M69 M78 N10 N53 N58 N80 O80 P30 P31 P65 P66 Q28 Q57 R26 R41 R49 R77 R8 S012 S2 S68 S79 T22 T2
8 U58 U76 V040 V34 V97 W48 W5 W69 X38 X46 X89 X93 Y036 Y49 Y66 Y87 Y88 Z69

Greedy Strategy:
Average Time: 0.045 ms
Sorted Data:
A30 B48 C22 C34 C34 C5 D17 D67 E78 E94 F08 F10 F11 F81 F91 G013 G15 G74 G77 G84 H18 H35 I88 J5 K51 K73
 L37 L96 M69 M78 N10 N53 N58 N80 O80 P30 P31 P65 P66 Q28 Q57 R26 R41 R49 R77 R8 S012 S2 S68 S79 T22 T2
8 U58 U76 V040 V34 V97 W48 W5 W69 X38 X46 X89 X93 Y036 Y49 Y66 Y87 Y88 Z69

Bubble Strategy:
Average Time: 0.055 ms
Sorted Data:
A30 B48 C22 C34 C34 C5 D17 D67 E78 E94 F08 F10 F11 F81 F91 G013 G15 G74 G77 G84 H18 H35 I88 J5 K51 K73
 L37 L96 M69 M78 N10 N53 N58 N80 O80 P30 P31 P65 P66 Q28 Q57 R26 R41 R49 R77 R8 S012 S2 S68 S79 T22 T2
8 U58 U76 V040 V34 V97 W48 W5 W69 X38 X46 X89 X93 Y036 Y49 Y66 Y87 Y88 Z69

Insertion Strategy:
Average Time: 0.029 ms
Sorted Data:
A30 B48 C22 C34 C34 C5 D17 D67 E78 E94 F08 F10 F11 F81 F91 G013 G15 G74 G77 G84 H18 H35 I88 J5 K51 K73
 L37 L96 M69 M78 N10 N53 N58 N80 O80 P30 P31 P65 P66 Q28 Q57 R26 R41 R49 R77 R8 S012 S2 S68 S79 T22 T2
8 U58 U76 V040 V34 V97 W48 W5 W69 X38 X46 X89 X93 Y036 Y49 Y66 Y87 Y88 Z69

Merge Strategy:
Average Time: 0.065 ms
Sorted Data:
A30 B48 C22 C34 C34 C5 D17 D67 E78 E94 F08 F10 F11 F81 F91 G013 G15 G74 G77 G84 H18 H35 I88 J5 K51 K73
 L37 L96 M69 M78 N10 N53 N58 N80 O80 P30 P31 P65 P66 Q28 Q57 R26 R41 R49 R77 R8 S012 S2 S68 S79 T22 T2
8 U58 U76 V040 V34 V97 W48 W5 W69 X38 X46 X89 X93 Y036 Y49 Y66 Y87 Y88 Z69


Process returned 0 (0x0)   execution time : 11.060 s
Press any key to continue.
```

```
Enter problem size: 80
Enter number of iterations: 800
Original Data:
P16 H2 Q69 G28 H36 U47 M29 E33 A39 Y99 L60 N72 L2 F5 D3 X95 F91 I83 R080 C15 V44 S82 C6 X57 G82 G95 B9
0 W084 K046 F81 N27 Q066 D76 U65 X5 W65 F84 N10 F10 O28 Z85 V60 S16 R62 T17 K99 J3 P93 R50 E80 P32 G74
 G48 X97 R17 P46 N14 R37 V28 Y044 S94 T73 M44 W9 C23 Y32 S50 Y74 Y58 C15 Q42 P31 E78 V9 I49 K063 E67 F
96 F31 M71


Results for 80 items, averaged over 800 iterations:

Brute Force Strategy:
Average Time: 0.131 ms
Sorted Data:
A39 B90 C15 C15 C23 C6 D3 D76 E33 E67 E78 E80 F10 F31 F5 F81 F84 F91 F96 G28 G48 G74 G82 G95 H2 H36 I4
9 I83 J3 K046 K063 K99 L2 L60 M29 M44 M71 N10 N14 N27 N72 O28 P16 P31 P32 P46 P93 Q066 Q42 Q69 R080 R1
7 R37 R50 R62 S16 S50 S82 S94 T17 T73 U47 U65 V28 V44 V60 V9 W084 W65 W9 X5 X57 X95 X97 Y044 Y32 Y58 Y
74 Y99 Z85

Greedy Strategy:
Average Time: 0.058 ms
Sorted Data:
A39 B90 C15 C15 C23 C6 D3 D76 E33 E67 E78 E80 F10 F31 F5 F81 F84 F91 F96 G28 G48 G74 G82 G95 H2 H36 I4
9 I83 J3 K046 K063 K99 L2 L60 M29 M44 M71 N10 N14 N27 N72 O28 P16 P31 P32 P46 P93 Q066 Q42 Q69 R080 R1
7 R37 R50 R62 S16 S50 S82 S94 T17 T73 U47 U65 V28 V44 V60 V9 W084 W65 W9 X5 X57 X95 X97 Y044 Y32 Y58 Y
74 Y99 Z85

Bubble Strategy:
Average Time: 0.077 ms
Sorted Data:
A39 B90 C15 C15 C23 C6 D3 D76 E33 E67 E78 E80 F10 F31 F5 F81 F84 F91 F96 G28 G48 G74 G82 G95 H2 H36 I4
9 I83 J3 K046 K063 K99 L2 L60 M29 M44 M71 N10 N14 N27 N72 O28 P16 P31 P32 P46 P93 Q066 Q42 Q69 R080 R1
7 R37 R50 R62 S16 S50 S82 S94 T17 T73 U47 U65 V28 V44 V60 V9 W084 W65 W9 X5 X57 X95 X97 Y044 Y32 Y58 Y
74 Y99 Z85

Insertion Strategy:
Average Time: 0.042 ms
Sorted Data:
A39 B90 C15 C15 C23 C6 D3 D76 E33 E67 E78 E80 F10 F31 F5 F81 F84 F91 F96 G28 G48 G74 G82 G95 H2 H36 I4
9 I83 J3 K046 K063 K99 L2 L60 M29 M44 M71 N10 N14 N27 N72 O28 P16 P31 P32 P46 P93 Q066 Q42 Q69 R080 R1
7 R37 R50 R62 S16 S50 S82 S94 T17 T73 U47 U65 V28 V44 V60 V9 W084 W65 W9 X5 X57 X95 X97 Y044 Y32 Y58 Y
74 Y99 Z85

Merge Strategy:
Average Time: 0.076 ms
Sorted Data:
A39 B90 C15 C15 C23 C6 D3 D76 E33 E67 E78 E80 F10 F31 F5 F81 F84 F91 F96 G28 G48 G74 G82 G95 H2 H36 I4
9 I83 J3 K046 K063 K99 L2 L60 M29 M44 M71 N10 N14 N27 N72 O28 P16 P31 P32 P46 P93 Q066 Q42 Q69 R080 R1
7 R37 R50 R62 S16 S50 S82 S94 T17 T73 U47 U65 V28 V44 V60 V9 W084 W65 W9 X5 X57 X95 X97 Y044 Y32 Y58 Y
74 Y99 Z85


Process returned 0 (0x0)   execution time : 3.856 s
```

```
Enter problem size: 90
Enter number of iterations: 900
Original Data:
P3 H90 Q40 G64 H10 U35 M74 E33 A18 Y08 L85 N21 L07 F74 D27 X92 F73 I78 R65 C18 V14 S29 C54 X54 G56 G69
 B10 W67 K46 F26 N86 Q9 D34 U63 X90 W18 F79 N69 F60 O57 Z51 V41 S76 R87 T80 K19 J55 P078 R28 E81 P86 G
38 G064 X68 R19 P33 N76 R56 V92 Y69 S26 T3 M70 W77 C3 Y11 S088 Y32 Y94 C60 Q89 P050 E69 V2 I46 K74 E79
 F27 F37 M62 Z77 N71 I32 M8 K57 K30 A23 S27 V49 W4


Results for 90 items, averaged over 900 iterations:

Brute Force Strategy:
Average Time: 0.160 ms
Sorted Data:
A18 A23 B10 C18 C3 C54 C60 D27 D34 E33 E69 E79 E81 F26 F27 F37 F60 F73 F74 F79 G064 G38 G56 G64 G69 H1
0 H90 I32 I46 I78 J55 K19 K30 K46 K57 K74 L07 L85 M62 M70 M74 M8 N21 N69 N71 N76 N86 O57 P050 P078 P3
P33 P86 Q40 Q89 Q9 R19 R28 R56 R65 R87 S088 S26 S27 S29 S76 T3 T80 U35 U63 V14 V2 V41 V49 V92 W18 W4 W
67 W77 X54 X68 X90 X92 Y08 Y11 Y32 Y69 Y94 Z51 Z77

Greedy Strategy:
Average Time: 0.069 ms
Sorted Data:
A18 A23 B10 C18 C3 C54 C60 D27 D34 E33 E69 E79 E81 F26 F27 F37 F60 F73 F74 F79 G064 G38 G56 G64 G69 H1
0 H90 I32 I46 I78 J55 K19 K30 K46 K57 K74 L07 L85 M62 M70 M74 M8 N21 N69 N71 N76 N86 O57 P050 P078 P3
P33 P86 Q40 Q89 Q9 R19 R28 R56 R65 R87 S088 S26 S27 S29 S76 T3 T80 U35 U63 V14 V2 V41 V49 V92 W18 W4 W
67 W77 X54 X68 X90 X92 Y08 Y11 Y32 Y69 Y94 Z51 Z77

Bubble Strategy:
Average Time: 0.100 ms
Sorted Data:
A18 A23 B10 C18 C3 C54 C60 D27 D34 E33 E69 E79 E81 F26 F27 F37 F60 F73 F74 F79 G064 G38 G56 G64 G69 H1
0 H90 I32 I46 I78 J55 K19 K30 K46 K57 K74 L07 L85 M62 M70 M74 M8 N21 N69 N71 N76 N86 O57 P050 P078 P3
P33 P86 Q40 Q89 Q9 R19 R28 R56 R65 R87 S088 S26 S27 S29 S76 T3 T80 U35 U63 V14 V2 V41 V49 V92 W18 W4 W
67 W77 X54 X68 X90 X92 Y08 Y11 Y32 Y69 Y94 Z51 Z77

Insertion Strategy:
Average Time: 0.052 ms
Sorted Data:
A18 A23 B10 C18 C3 C54 C60 D27 D34 E33 E69 E79 E81 F26 F27 F37 F60 F73 F74 F79 G064 G38 G56 G64 G69 H1
0 H90 I32 I46 I78 J55 K19 K30 K46 K57 K74 L07 L85 M62 M70 M74 M8 N21 N69 N71 N76 N86 O57 P050 P078 P3
P33 P86 Q40 Q89 Q9 R19 R28 R56 R65 R87 S088 S26 S27 S29 S76 T3 T80 U35 U63 V14 V2 V41 V49 V92 W18 W4 W
67 W77 X54 X68 X90 X92 Y08 Y11 Y32 Y69 Y94 Z51 Z77

Merge Strategy:
Average Time: 0.089 ms
Sorted Data:
A18 A23 B10 C18 C3 C54 C60 D27 D34 E33 E69 E79 E81 F26 F27 F37 F60 F73 F74 F79 G064 G38 G56 G64 G69 H1
0 H90 I32 I46 I78 J55 K19 K30 K46 K57 K74 L07 L85 M62 M70 M74 M8 N21 N69 N71 N76 N86 O57 P050 P078 P3
P33 P86 Q40 Q89 Q9 R19 R28 R56 R65 R87 S088 S26 S27 S29 S76 T3 T80 U35 U63 V14 V2 V41 V49 V92 W18 W4 W
67 W77 X54 X68 X90 X92 Y08 Y11 Y32 Y69 Y94 Z51 Z77


Process returned 0 (0x0)   execution time : 4.580 s
```

```
Enter problem size: 100
Enter number of iterations: 1000
Original Data:
P83 H8 Q78 G79 H35 U091 M41 E085 A34 Y59 L29 N40 L67 F17 D79 X016 F26 I41 R62 C59 V037 S80 C59 X43 G96
 G73 B062 W53 K060 F014 N4 Q76 D39 U55 X67 W56 F2 N6 F57 O73 Z48 V41 S81 R20 T25 K2 J87 P77 R37 E038 P
9 G53 G93 X72 R59 P98 N48 R7 V51 Y45 S69 T094 M21 W072 C57 Y087 S24 Y99 Y17 C67 Q54 P90 E79 V40 I064 K
77 E98 F6 F73 M91 Z73 N80 I25 M71 K9 K83 A39 S56 V27 W74 S62 R39 E045 N054 Z68 K73 Y77 C81 X28 F85


Results for 100 items, averaged over 1000 iterations:

Brute Force Strategy:
Average Time: 0.205 ms
Sorted Data:
A34 A39 B062 C57 C59 C59 C67 C81 D39 D79 E038 E045 E085 E79 E98 F014 F17 F2 F26 F57 F6 F73 F85 G53 G73
 G79 G93 G96 H35 H8 I064 I25 I41 J87 K060 K2 K73 K77 K83 K9 L29 L67 M21 M41 M71 M91 N054 N4 N40 N48 N6
 N80 O73 P77 P83 P9 P90 P98 Q54 Q76 Q78 R20 R37 R39 R59 R62 R7 S24 S56 S62 S69 S80 S81 T094 T25 U091 U
55 V037 V27 V40 V41 V51 W072 W53 W56 W74 X016 X28 X43 X67 X72 Y087 Y17 Y45 Y59 Y77 Y99 Z48 Z68 Z73

Greedy Strategy:
Average Time: 0.091 ms
Sorted Data:
A34 A39 B062 C57 C59 C59 C67 C81 D39 D79 E038 E045 E085 E79 E98 F014 F17 F2 F26 F57 F6 F73 F85 G53 G73
 G79 G93 G96 H35 H8 I064 I25 I41 J87 K060 K2 K73 K77 K83 K9 L29 L67 M21 M41 M71 M91 N054 N4 N40 N48 N6
 N80 O73 P77 P83 P9 P90 P98 Q54 Q76 Q78 R20 R37 R39 R59 R62 R7 S24 S56 S62 S69 S80 S81 T094 T25 U091 U
55 V037 V27 V40 V41 V51 W072 W53 W56 W74 X016 X28 X43 X67 X72 Y087 Y17 Y45 Y59 Y77 Y99 Z48 Z68 Z73

Bubble Strategy:
Average Time: 0.142 ms
Sorted Data:
A34 A39 B062 C57 C59 C59 C67 C81 D39 D79 E038 E045 E085 E79 E98 F014 F17 F2 F26 F57 F6 F73 F85 G53 G73
 G79 G93 G96 H35 H8 I064 I25 I41 J87 K060 K2 K73 K77 K83 K9 L29 L67 M21 M41 M71 M91 N054 N4 N40 N48 N6
 N80 O73 P77 P83 P9 P90 P98 Q54 Q76 Q78 R20 R37 R39 R59 R62 R7 S24 S56 S62 S69 S80 S81 T094 T25 U091 U
55 V037 V27 V40 V41 V51 W072 W53 W56 W74 X016 X28 X43 X67 X72 Y087 Y17 Y45 Y59 Y77 Y99 Z48 Z68 Z73

Insertion Strategy:
Average Time: 0.071 ms
Sorted Data:
A34 A39 B062 C57 C59 C59 C67 C81 D39 D79 E038 E045 E085 E79 E98 F014 F17 F2 F26 F57 F6 F73 F85 G53 G73
 G79 G93 G96 H35 H8 I064 I25 I41 J87 K060 K2 K73 K77 K83 K9 L29 L67 M21 M41 M71 M91 N054 N4 N40 N48 N6
 N80 O73 P77 P83 P9 P90 P98 Q54 Q76 Q78 R20 R37 R39 R59 R62 R7 S24 S56 S62 S69 S80 S81 T094 T25 U091 U
55 V037 V27 V40 V41 V51 W072 W53 W56 W74 X016 X28 X43 X67 X72 Y087 Y17 Y45 Y59 Y77 Y99 Z48 Z68 Z73

Merge Strategy:
Average Time: 0.111 ms
Sorted Data:
A34 A39 B062 C57 C59 C59 C67 C81 D39 D79 E038 E045 E085 E79 E98 F014 F17 F2 F26 F57 F6 F73 F85 G53 G73
 G79 G93 G96 H35 H8 I064 I25 I41 J87 K060 K2 K73 K77 K83 K9 L29 L67 M21 M41 M71 M91 N054 N4 N40 N48 N6
 N80 O73 P77 P83 P9 P90 P98 Q54 Q76 Q78 R20 R37 R39 R59 R62 R7 S24 S56 S62 S69 S80 S81 T094 T25 U091 U
55 V037 V27 V40 V41 V51 W072 W53 W56 W74 X016 X28 X43 X67 X72 Y087 Y17 Y45 Y59 Y77 Y99 Z48 Z68 Z73


Process returned 0 (0x0)   execution time : 4.713 s
```

5. Hasil Pengujian

   Setelah pengujian dilakukan, waktu eksekusi untuk masing-masing algoritma dapat dicatat dalam tabel dengan kolom sebagai berikut:

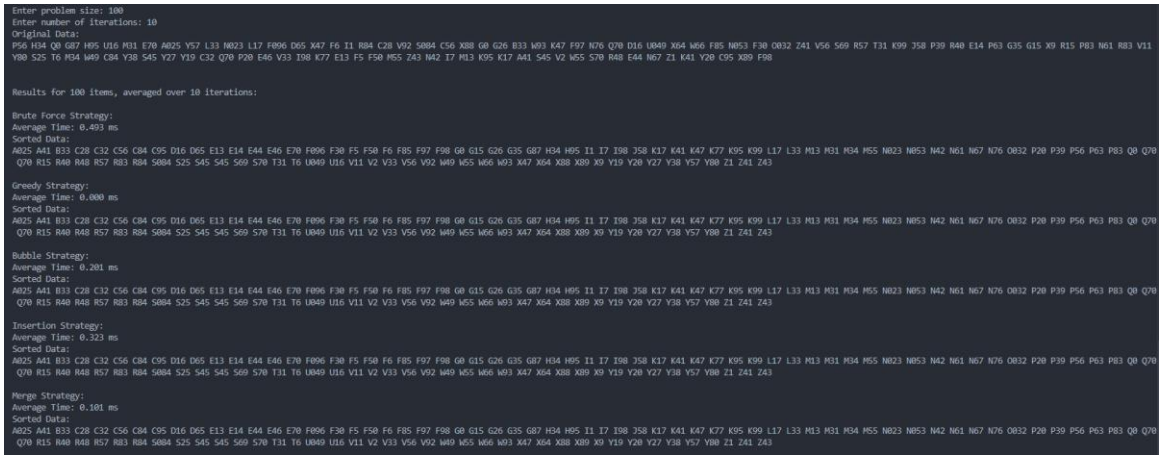
   Keterangan:

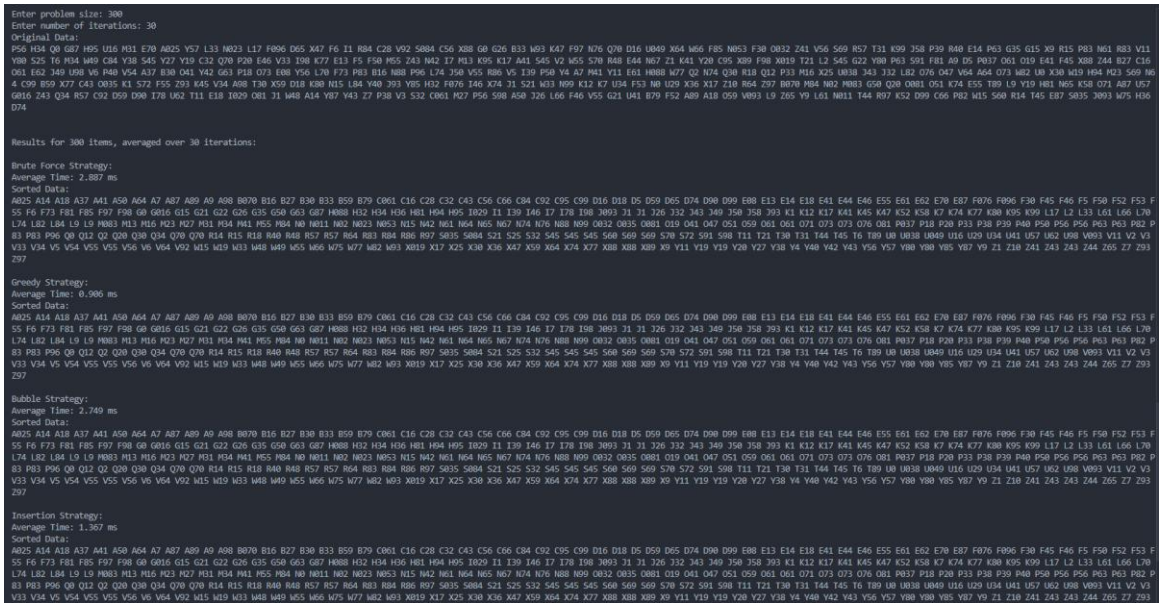
   n = jumlah inputan

Tabel Hasil Perbandingan Pada Proses Problem Size

No	Problem Size (n)	Iteration	Brute Force	Greedy	Bubble	Insertion	Merge
1	10	100	0.007	0.002	0.002	0.002	0.010
2	20	200	0.016	0.005	0.006	0.005	0.020
3	30	300	0.027	0.010	0.012	0.008	0.029
4	40	400	0.040	0.016	0.018	0.012	0.035
5	50	500	0.58	0.024	0.029	0.018	0.048
6	60	600	0.79	0.033	0.039	0.023	0.054
7	70	700	0.105	0.045	0.055	0.029	0.065
8	80	800	0.131	0.058	0.077	0.042	0.076
9	90	900	0.160	0.069	0.100	0.052	0.089
0	100	1000	0.205	0.091	0.142	0.071	0.111

Lampiran Proses Iteration





```
Enter problem size: 400
Enter number of iterations: 40
Original Data:
P56 H34 Q0 G87 H95 U16 M31 E70 A025 Y57 L33 N023 L17 F096 D65 X47 F6 I1 R84 C28 V92 S084 C56 X88 G0 G26 B33 W03 K47 F97 N76 Q70 D16 U049 X64 W66 F85 N053 F30 O032 Z41 V56 S69 R57 T31 K99 J58 P39 R40 E14 P63 G35 G15 X9 R15 P83 N61 R83 V11 Y80 S25 T6 M34 W49 C84 Y38 S45 Y27 Y19 C32 Q70 P20 E46 V33 I98 K77 E13 F5 F50 M55 Z43 W42 I7 M13 K95 K17 A41 S45 V2 W55 S70 R48 E44 N67 Z1 K41 Y20 C95 X89 F98 X019 T21 L2 S45 G22 Y80 P63 S91 F81 A9 D5 P037 O61 O19 E41 F45 X88 Z44 B27 C16 O61 E62 J49 U98 V6 P40 V54 A37 B30 O41 Y42 G63 P18 O73 E08 Y56 L70 F73 P83 B16 N88 P96 L74 J58 V55 R86 V5 I39 P50 Y4 A7 M41 Y11 E61 H088 W77 Q2 N74 Q30 R10 Q12 P33 M16 X25 U038 J43 J32 L82 O76 O47 V64 A64 O73 W82 U0 X30 W19 H94 M23 S69 N6 4 C99 B59 X27 C43 O035 K1 S72 F55 Z03 K45 V34 A00 T30 X59 D18 K80 N15 L84 Y40 J93 Y05 H32 F076 I46 X74 J1 S21 W33 N00 K12 K7 U34 F53 N0 U29 X36 X17 Z10 R64 Z97 B070 M94 W02 M083 G50 Q00 O001 O51 K74 E55 T89 L9 Y19 H81 N65 K58 O71 A07 U57 G016 Z43 Q34 R57 C92 D59 D90 I78 U62 T11 E18 I029 O81 J1 W48 A14 Y67 Y43 Z7 P38 V3 S32 O861 M27 P56 S98 A50 J26 L66 F46 V55 G21 U41 B79 T82 A89 A18 O59 W093 L9 Z65 Y9 L61 H011 T44 R97 K52 D99 C66 P82 W15 S60 R14 T45 E87 S035 J093 W75 H56 D74 I56 Z071 C33 O025 B22 Z15 C066 N0 F91 W6 L016 Q14 I15 J59 T13 V65 D81 W49 V17 X3 H27 R99 C5 B13 L53 D87 V1 G9 Y5 L53 W73 G88 B94 U20 S58 B34 M031 B028 Q29 R99 X13 T26 L7 H5 C07 S2 M83 P17 X90 O12 H19 G42 M0 G38 N076 K25 E56 U91 F67 D2 9 X0 O15 T28 O93 G95 B82 G98 X17 P40 E41 Y64 A0 N71 F018 E60 T40 C0 U86 K94 E91 P23 Z61 S62 H81 K21 L9 J018 U86 G56 G25 G67 E87 K32 J04 D32 Q9 Z34 J11 E55 N61


Results for 400 items, averaged over 40 iterations:

Brute Force Strategy:
Average Time: 5.005 ms
Sorted Data:
A0 A025 A14 A18 A37 A41 A50 A64 A7 A87 A89 A9 A98 B028 B070 B13 B16 B22 B27 B30 B33 B34 B59 B79 B82 B94 C0 C061 C066 C07 C16 C28 C32 C33 C43 C5 C56 C66 C84 C92 C95 C99 D16 D18 D29 D32 D5 D59 D65 D74 D81 D87 D90 D99 E08 E13 E14 E18 E41 E41
 E44 E46 E55 E5S E56 E60 E61 E62 E70 E87 E87 E91 F018 F076 F096 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F85 F91 F97 F98 G0 G016 G25 G27 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H27 H32 H34 H36 H5 H81 H81 H
 94 H95 I029 I1 I15 I39 I46 I56 I7 I78 I98 J018 J04 J093 J1 J1 J11 J26 J32 J43 J49 J50 J58 J59 J93 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74 K77 K80 K94 K95 K99 L016 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M0
 83 M13 M16 M23 M27 M31 M34 M41 M55 M83 M84 N0 N0 N011 N02 N023 N053 N076 N15 N42 N61 N61 N64 N65 N67 N71 N74 N76 N88 N99 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O59 O61 O61 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P33 P38 P39
 P40 P40 P50 P56 P56 P63 P63 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q30 Q34 Q70 Q70 Q9 R14 R15 R18 R40 R48 R57 R57 R64 R83 R84 R86 R97 R99 R99 S035 S084 S2 S21 S25 S32 S45 S45 S55 S58 S60 S62 S69 S69 S70 S72 S91 S98 T11 T13 T21 T26 T28 T30 T31
 T40 T44 T45 T6 T89 U0 U038 U049 U16 U20 U29 U34 U41 U57 U62 U86 U86 U91 U98 V093 V1 V11 V17 V2 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V92 W15 W19 W33 W48 W49 W49 W55 W6 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59
 X64 X74 X77 X88 X88 X89 X9 X90 Y11 Y19 Y19 Y20 Y27 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y64 Y80 Y80 Y85 Y87 Y9 Z071 Z1 Z10 Z15 Z34 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z93 Z97

Greedy Strategy:
Average Time: 1.657 ms
Sorted Data:
A0 A025 A14 A18 A37 A41 A50 A64 A7 A87 A89 A9 A98 B028 B070 B13 B16 B22 B27 B30 B33 B34 B59 B79 B82 B94 C0 C061 C066 C07 C16 C28 C32 C33 C43 C5 C56 C66 C84 C92 C95 C99 D16 D18 D29 D32 D5 D59 D65 D74 D81 D87 D90 D99 E08 E13 E14 E18 E41 E41
 E44 E46 E55 E5S E56 E60 E61 E62 E70 E87 E87 E91 F018 F076 F096 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F85 F91 F97 F98 G0 G016 G25 G27 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H27 H32 H34 H36 H5 H81 H81 H
 94 H95 I029 I1 I15 I39 I46 I56 I7 I78 I98 J018 J04 J093 J1 J1 J11 J26 J32 J43 J49 J50 J58 J59 J93 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74 K77 K80 K94 K95 K99 L016 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M0
 83 M13 M16 M23 M27 M31 M34 M41 M55 M83 M84 N0 N0 N011 N02 N023 N053 N076 N15 N42 N61 N61 N64 N65 N67 N71 N74 N76 N88 N99 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O59 O61 O61 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P33 P38 P39
 P40 P40 P50 P56 P56 P63 P63 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q30 Q34 Q70 Q70 Q9 R14 R15 R18 R40 R48 R57 R57 R64 R83 R84 R86 R97 R99 R99 S035 S084 S2 S21 S25 S32 S45 S45 S55 S58 S60 S62 S69 S69 S70 S72 S91 S98 T11 T13 T21 T26 T28 T30 T31
 T40 T44 T45 T6 T89 U0 U038 U049 U16 U20 U29 U34 U41 U57 U62 U86 U86 U91 U98 V093 V1 V11 V17 V2 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V92 W15 W19 W33 W48 W49 W49 W55 W6 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59
 X64 X74 X77 X88 X88 X89 X9 X90 Y11 Y19 Y19 Y20 Y27 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y64 Y80 Y80 Y85 Y87 Y9 Z071 Z1 Z10 Z15 Z34 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z93 Z97

Bubble Strategy:
Average Time: 4.748 ms
Sorted Data:
A0 A025 A14 A18 A37 A41 A50 A64 A7 A87 A89 A9 A98 B028 B070 B13 B16 B22 B27 B30 B33 B34 B59 B79 B82 B94 C0 C061 C066 C07 C16 C28 C32 C33 C43 C5 C56 C66 C84 C92 C95 C99 D16 D18 D29 D32 D5 D59 D65 D74 D81 D87 D90 D99 E08 E13 E14 E18 E41 E41
 E44 E46 E55 E5S E56 E60 E61 E62 E70 E87 E87 E91 F018 F076 F096 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F85 F91 F97 F98 G0 G016 G25 G27 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H27 H32 H34 H36 H5 H81 H81 H
 94 H95 I029 I1 I15 I39 I46 I56 I7 I78 I98 J018 J04 J093 J1 J1 J11 J26 J32 J43 J49 J50 J58 J59 J93 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74 K77 K80 K94 K95 K99 L016 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M0
 83 M13 M16 M23 M27 M31 M34 M41 M55 M83 M84 N0 N0 N011 N02 N023 N053 N076 N15 N42 N61 N61 N64 N65 N67 N71 N74 N76 N88 N99 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O59 O61 O61 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P33 P38 P39
 P40 P40 P50 P56 P56 P63 P63 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q30 Q34 Q70 Q70 Q9 R14 R15 R18 R40 R48 R57 R57 R64 R83 R84 R86 R97 R99 R99 S035 S084 S2 S21 S25 S32 S45 S45 S55 S58 S60 S62 S69 S69 S70 S72 S91 S98 T11 T13 T21 T26 T28 T30 T31
 T40 T44 T45 T6 T89 U0 U038 U049 U16 U20 U29 U34 U41 U57 U62 U86 U86 U91 U98 V093 V1 V11 V17 V2 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V92 W15 W19 W33 W48 W49 W49 W55 W6 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59
 X64 X74 X77 X88 X88 X89 X9 X90 Y11 Y19 Y19 Y20 Y27 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y64 Y80 Y80 Y85 Y87 Y9 Z071 Z1 Z10 Z15 Z34 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z93 Z97

Insertion Strategy:
Average Time: 2.631 ms
Sorted Data:
A0 A025 A14 A18 A37 A41 A50 A64 A7 A87 A89 A9 A98 B028 B070 B13 B16 B22 B27 B30 B33 B34 B59 B79 B82 B94 C0 C061 C066 C07 C16 C28 C32 C33 C43 C5 C56 C66 C84 C92 C95 C99 D16 D18 D29 D32 D5 D59 D65 D74 D81 D87 D90 D99 E08 E13 E14 E18 E41 E41
 E44 E46 E55 E5S E56 E60 E61 E62 E70 E87 E87 E91 F018 F076 F096 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F85 F91 F97 F98 G0 G016 G25 G27 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H27 H32 H34 H36 H5 H81 H81 H
 94 H95 I029 I1 I15 I39 I46 I56 I7 I78 I98 J018 J04 J093 J1 J1 J11 J26 J32 J43 J49 J50 J58 J59 J93 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74 K77 K80 K94 K95 K99 L016 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M0
 83 M13 M16 M23 M27 M31 M34 M41 M55 M83 M84 N0 N0 N011 N02 N023 N053 N076 N15 N42 N61 N61 N64 N65 N67 N71 N74 N76 N88 N99 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O59 O61 O61 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P33 P38 P39
 P40 P40 P50 P56 P56 P63 P63 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q30 Q34 Q70 Q70 Q9 R14 R15 R18 R40 R48 R57 R57 R64 R83 R84 R86 R97 R99 R99 S035 S084 S2 S21 S25 S32 S45 S45 S55 S58 S60 S62 S69 S69 S70 S72 S91 S98 T11 T13 T21 T26 T28 T30 T31
 T40 T44 T45 T6 T89 U0 U038 U049 U16 U20 U29 U34 U41 U57 U62 U86 U86 U91 U98 V093 V1 V11 V17 V2 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V92 W15 W19 W33 W48 W49 W49 W55 W6 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59
 X64 X74 X77 X88 X88 X89 X9 X90 Y11 Y19 Y19 Y20 Y27 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y64 Y80 Y80 Y85 Y87 Y9 Z071 Z1 Z10 Z15 Z34 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z93 Z97
```

Ln 16, Col 35 Spaces: 4 UTF-8 CRLF {} C++ Go Live Quokka Win32 Go Live

```
Enter problem size: 500
Enter number of iterations: 50
Original Data:
P56 H34 Q0 G87 H95 U16 M31 E70 A025 Y57 L33 N023 L17 F096 D65 X47 F6 I1 R84 C28 V92 S084 C56 X88 G0 G26 B33 W03 K47 F97 N76 Q70 D16 U049 X64 W66 F85 N053 F30 O032 Z41 V56 S69 R57 T31 K99 J58 P39 R40 E14 P63 G35 G15 X9 R15 P83 N61 R83 V11 Y80 S25 T6 M34 W49 C84 Y
38 S45 Y27 Y19 C32 Q70 P20 E46 V33 I98 K77 E13 F5 F50 M55 Z43 W42 I7 M13 K95 K17 A41 S45 V2 W55 S70 R48 E44 N67 Z1 K41 Y20 C95 X89 F98 X019 T21 L2 S45 G22 Y80 P63 S91 F81 A9 D5 P037 O61 O19 E41 F45 X88 Z44 B27 C16 O61 E62 J49 U98 V6 P40 V54 A37 B30 O41 Y42 G63 P
18 O73 E08 Y56 L70 F73 P83 B16 N88 P96 L74 J58 V55 R86 V5 I39 P50 Y4 A7 M41 Y11 E61 H088 W77 Q2 N74 Q30 R10 Q12 P33 M16 X25 U038 J43 J32 L82 O76 O47 V64 A64 O73 W82 U0 X30 W19 H94 M23 S69 N6 4 C99 B59 X27 C43 O035 K1 S72 F55 Z03 K45 V34 A00 T30 X59 D18 K80 N15 L8
4 Y40 J93 Y05 H32 F076 I46 X74 J1 S21 W33 N00 K12 K7 U34 F53 N0 U29 X36 X17 Z10 R64 Z97 B070 M94 W02 M083 G50 Q00 O001 O51 K74 E55 T89 L9 Y19 H81 N65 K58 O71 A07 U57 G016 Z43 Q34 R57 C92 D59 D90 I78 U62 T11 E18 I029 O81 J1 W48 A14 Y67 Y43 Z7 P38 V3 S32 O861 M27
P56 S98 A50 J26 L66 F46 V55 G21 U41 B79 F52 A89 A18 O59 W093 L9 Z65 Y9 L61 H011 T44 R97 K52 D99 C66 P82 W15 S60 R14 T45 E87 S035 J093 W75 H36 D74 I56 Z071 C33 O025 B22 Z15 C066 N0 F91 W6 L016 Q14 I15 J59 T13 V65 D81 W49 V17 X3 H27 R99 C5 B13 L53 D87 V1 G9 Y5 L53
W73 G88 B94 U20 S58 B34 M031 B028 Q29 R99 X13 T26 L7 H5 C07 S2 M83 P17 X90 O12 H19 G42 M0 G38 N076 K25 E56 U91 F67 D29 X0 O15 T28 O93 G95 B82 G98 X17 P40 E41 Y64 A0 N71 F018 E60 T40 C0 U86 K94 E91 P23 Z61 S62 H81 K21 L9 J018 U86 G56 G25 G67 E87 K32 J04 D32 Q9 Z
34 J11 E55 N61 P42 E50 V27 Q26 G3 X95 E79 E60 P28 J84 S62 R55 D2 Z41 J90 A26 Z38 U37 J2 L17 L12 C41 H84 H26 B77 F82 Q51 M60 X97 I38 M19 W62 Z89 O56 B44 I54 W64 Y58 B15 X8 D27 U40 U66 N63 F27 S62 K9 S3 R47 S78 R71 T12 E38 K1 M49 Q74 D036 C0 Y58 Z93 384 E86 E3 U02
H88 M33 S90 R9 Q47 C58 O60 Z012 I45 J67 I38 P41 F22 I99 O022 M97 E61 E070 D0 D82 P68 S83 Z22 R65 N851 A56 V74 Y36 M88 H77 T41 A82 T75 B20 D37 Z26


Results for 500 items, averaged over 50 iterations:

Brute Force Strategy:
Average Time: 7.575 ms
Sorted Data:
A0 A025 A14 A18 A26 A37 A41 A50 A56 A64 A7 A82 A87 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B30 B33 B34 B44 B59 B77 B79 B82 B94 C0 C0 C061 C066 C07 C16 C28 C32 C33 C41 C43 C5 C56 C58 C66 C84 C92 C95 C99 D0 D036 D16 D18 D2 D27 D29 D32 D37 D5 D59 D65 D74 D81 D
82 D87 D90 D99 E070 E08 E13 E14 E18 E3 E38 E41 E41 E44 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67
G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H5 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I56 I7 I78 I79 I98 I99 J018 J04 J093 J1 J1 J11 J2 J26 J32 J43 J49 J50 J58 J59 J67 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74
K77 K80 K9 K94 K95 K97 K99 L016 L12 L17 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M083 M13 M16 M19 M23 M27 M31 M33 M34 M41 M49 M50 M55 M77 M83 M84 M88 N0 N0 N011 N02 N023 N051 N053 N076 N15 N42 N61 N61 N63 N64 N65 N67 N71 N74 N76 N88 N97 N9
9 O022 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O56 O59 O60 O63 O81 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P28 P33 P38 P39 P40 P40 P41 P42 P50 P56 P56 P63 P63 P68 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q26 Q30 Q34 Q47 Q51 Q70 Q70 Q74 Q9 R14 R15 R18 R40 R
47 R48 R55 R57 R57 R64 R65 R71 R83 R84 R86 R9 R97 R99 R99 S035 S084 S2 S21 S25 S3 S32 S45 S45 S45 S58 S60 S62 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S98 T11 T12 T13 T21 T26 T28 T30 T31 T40 T41 T44 T45 T6 T75 T89 U0 U038 U049 U16 U20 U29 U34 U37 U40 U41 U57 U62
U66 U86 U86 U91 U92 U98 V093 V1 V11 V17 V2 V27 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V74 V92 W15 W19 W33 W48 W49 W49 W55 W5 W62 W64 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59 X64 X74 X77 X8 X88 X88 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y3
6 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y58 Y58 Y64 Y80 Y80 Y85 Y87 Y9 Z012 Z071 Z1 Z10 Z15 Z22 Z26 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z89 Z93 Z93 Z97

Greedy Strategy:
Average Time: 2.608 ms
Sorted Data:
A0 A025 A14 A18 A26 A37 A41 A50 A56 A64 A7 A82 A87 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B30 B33 B34 B44 B59 B77 B79 B82 B94 C0 C0 C061 C066 C07 C16 C28 C32 C33 C41 C43 C5 C56 C58 C66 C84 C92 C95 C99 D0 D036 D16 D18 D2 D27 D29 D32 D37 D5 D59 D65 D74 D81 D
82 D87 D90 D99 E070 E08 E13 E14 E18 E3 E38 E41 E41 E44 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67
G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H5 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I56 I7 I78 I79 I98 I99 J018 J04 J093 J1 J1 J11 J2 J26 J32 J43 J49 J50 J58 J59 J67 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74
K77 K80 K9 K94 K95 K97 K99 L016 L12 L17 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M083 M13 M16 M19 M23 M27 M31 M33 M34 M41 M49 M50 M55 M77 M83 M84 M88 N0 N0 N011 N02 N023 N051 N053 N076 N15 N42 N61 N61 N63 N64 N65 N67 N71 N74 N76 N88 N97 N9
9 O022 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O56 O59 O60 O63 O81 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P28 P33 P38 P39 P40 P40 P41 P42 P50 P56 P56 P63 P63 P68 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q26 Q30 Q34 Q47 Q51 Q70 Q70 Q74 Q9 R14 R15 R18 R40 R
47 R48 R55 R57 R57 R64 R65 R71 R83 R84 R86 R9 R97 R99 R99 S035 S084 S2 S21 S25 S3 S32 S45 S45 S45 S58 S60 S62 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S98 T11 T12 T13 T21 T26 T28 T30 T31 T40 T41 T44 T45 T6 T75 T89 U0 U038 U049 U16 U20 U29 U34 U37 U40 U41 U57 U62
U66 U86 U86 U91 U92 U98 V093 V1 V11 V17 V2 V27 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V74 V92 W15 W19 W33 W48 W49 W49 W55 W5 W62 W64 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59 X64 X74 X77 X8 X88 X88 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y3
6 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y58 Y58 Y64 Y80 Y80 Y85 Y87 Y9 Z012 Z071 Z1 Z10 Z15 Z22 Z26 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z89 Z93 Z93 Z97

Bubble Strategy:
Average Time: 7.712 ms
Sorted Data:
A0 A025 A14 A18 A26 A37 A41 A50 A56 A64 A7 A82 A87 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B30 B33 B34 B44 B59 B77 B79 B82 B94 C0 C0 C061 C066 C07 C16 C28 C32 C33 C41 C43 C5 C56 C58 C66 C84 C92 C95 C99 D0 D036 D16 D18 D2 D27 D29 D32 D37 D5 D59 D65 D74 D81 D
82 D87 D90 D99 E070 E08 E13 E14 E18 E3 E38 E41 E41 E44 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67
G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H5 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I56 I7 I78 I79 I98 I99 J018 J04 J093 J1 J1 J11 J2 J26 J32 J43 J49 J50 J58 J59 J67 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K7 K74
K77 K80 K9 K94 K95 K97 K99 L016 L12 L17 L17 L2 L33 L53 L53 L61 L66 L7 L70 L74 L82 L84 L9 L9 L9 M0 M031 M083 M13 M16 M19 M23 M27 M31 M33 M34 M41 M49 M50 M55 M77 M83 M84 M88 N0 N0 N011 N02 N023 N051 N053 N076 N15 N42 N61 N61 N63 N64 N65 N67 N71 N74 N76 N88 N97 N9
9 O022 O025 O032 O035 O081 O12 O15 O19 O29 O41 O47 O51 O56 O59 O60 O63 O81 O71 O73 O73 O76 O81 O93 P037 P17 P18 P20 P23 P28 P33 P38 P39 P40 P40 P41 P42 P50 P56 P56 P63 P63 P68 P82 P83 P83 P96 Q0 Q12 Q14 Q2 Q20 Q26 Q30 Q34 Q47 Q51 Q70 Q70 Q74 Q9 R14 R15 R18 R40 R
47 R48 R55 R57 R57 R64 R65 R71 R83 R84 R86 R9 R97 R99 R99 S035 S084 S2 S21 S25 S3 S32 S45 S45 S45 S58 S60 S62 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S98 T11 T12 T13 T21 T26 T28 T30 T31 T40 T41 T44 T45 T6 T75 T89 U0 U038 U049 U16 U20 U29 U34 U37 U40 U41 U57 U62
U66 U86 U86 U91 U92 U98 V093 V1 V11 V17 V2 V27 V3 V33 V34 V5 V54 V55 V55 V56 V6 V64 V65 V74 V92 W15 W19 W33 W48 W49 W49 W55 W5 W62 W64 W66 W73 W75 W77 W82 W93 X0 X019 X13 X17 X17 X25 X3 X30 X36 X47 X59 X64 X74 X77 X8 X88 X88 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y3
6 Y38 Y4 Y40 Y42 Y43 Y5 Y56 Y57 Y58 Y58 Y64 Y80 Y80 Y85 Y87 Y9 Z012 Z071 Z1 Z10 Z15 Z22 Z26 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z61 Z65 Z7 Z89 Z93 Z93 Z97

Insertion Strategy:
Average Time: 3.848 ms
Sorted Data:
A0 A025 A14 A18 A26 A37 A41 A50 A56 A64 A7 A82 A87 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B30 B33 B34 B44 B59 B77 B79 B82 B94 C0 C0 C061 C066 C07 C16 C28 C32 C33 C41 C43 C5 C56 C58 C66 C84 C92 C95 C99 D0 D036 D16 D18 D2 D27 D29 D32 D37 D5 D59 D65 D74 D81 D
```

⚠ 0 ⚠ 0 Ln 16, Col 35 Spaces: 4 UTF-8 CRLF {} C++ Go Live Quokka Win32 Go Live

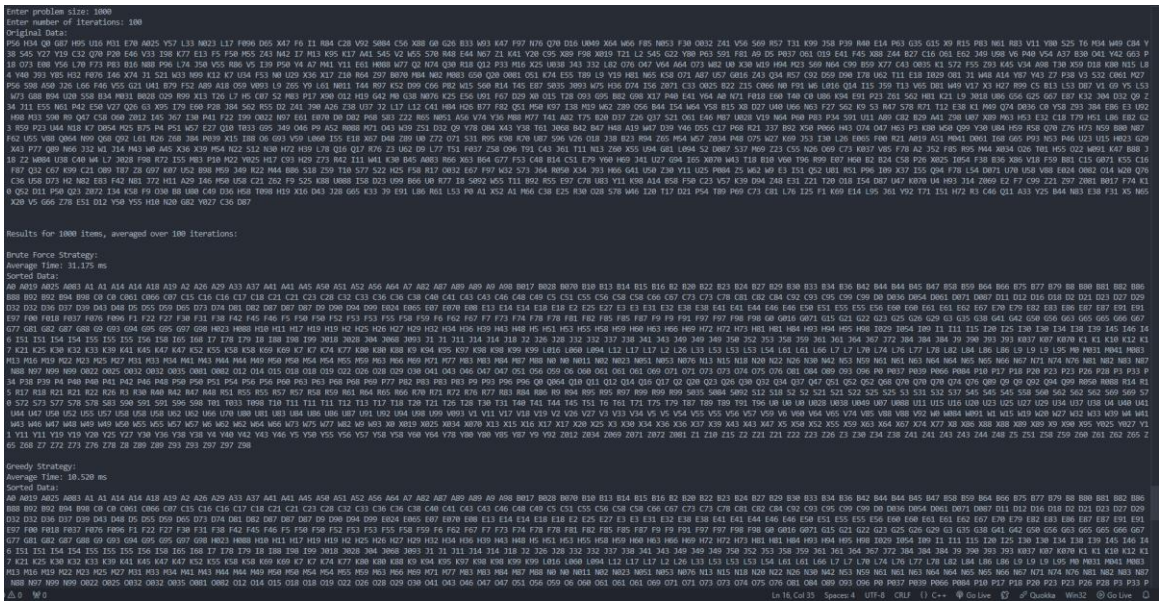
```
Enter problem size: 600
Enter number of iterations: 60
Original Data:
P56 H34 Q0 G87 H95 U16 M31 E70 A025 Y57 L33 N023 L17 F096 D65 X47 F6 I1 R84 C28 V92 S084 C56 X88 G0 G26 B33 W93 K47 F97 N76 Q70 D16 U049 X64 W66 F85 N053 F30 O032 Z41 V56 S69 R57 T31 K99 J58 P39 A40 E14 P63 G35 G15 X9 R15 P83 N61 R83 V11 Y80 S25 T6 M34 W49 C84 Y
38 S45 Y27 Y19 C32 Q70 P20 E46 V33 I98 K77 E13 F5 F50 M55 Z43 W42 I7 M13 K95 K17 A41 S45 V2 W55 S70 B48 E44 N67 Z1 K41 Y20 C95 X89 F98 X019 T21 L2 S45 G22 Y80 P63 S91 F81 A9 D5 P037 O61 O19 E41 F45 X88 Z44 B27 C16 O61 E62 J49 U88 V6 P40 V54 A37 E30 O41 Y42 G63 P
18 O73 E08 Y56 L70 F73 P83 B16 N88 P96 L74 J50 V55 R86 V5 I39 P50 Y4 A7 M41 Y11 E61 H088 W77 Q2 N74 Q38 R18 Q12 P33 M16 X25 U038 J43 T32 L82 O76 O47 V64 A64 O73 W82 U0 X30 W19 H94 M23 S69 N64 C99 B59 X77 C43 O035 K1 S72 F55 Z93 K45 V34 A08 T30 X59 D18 K80 N15 L8
4 Y40 J93 Y85 H32 F076 I46 X74 J1 S21 W33 N99 K12 K7 U34 F53 N0 U29 X36 X17 Z10 R64 Z97 B070 M84 N02 M083 G50 Q20 O881 O51 K74 E55 T89 L9 Y19 H81 N65 K58 O71 A87 U57 G016 Z43 Q34 R57 C92 D59 D90 I78 U62 T11 E18 I029 O81 J1 W48 A14 Y87 Y43 Z7 P38 V3 S32 C061 M27
P56 S98 A50 J26 L66 F46 V55 G21 U41 B79 F52 A89 A10 O59 V093 L9 Z65 Y9 L61 N011 T44 R97 K52 D99 C66 P82 W15 S60 R14 T45 E87 S035 J093 W75 H36 O74 I56 Z071 C33 O025 B22 Z15 C066 N0 F91 W6 U016 Q14 I15 J59 T13 V65 D81 W49 V17 X3 H27 R99 C5 B113 L53 D87 V1 G0 Y5 L53
N73 G88 B94 U20 S58 B34 M031 B028 O29 R99 X13 T26 L7 H5 C87 S2 M83 P17 X90 O12 H59 G42 M8 G38 N076 K25 E56 U91 F67 D29 X0 O15 T28 O93 G95 B82 G98 X17 P40 E41 V64 A0 N71 F018 E60 T40 C0 U86 K94 E91 P23 Z61 S62 H81 K21 L9 J018 U86 G56 G25 G67 E87 K32 J04 D32 Q0 Z
34 J11 E55 N61 P42 E50 V27 Q26 G3 X95 I79 E60 P28 J84 S62 R55 D2 Z41 J90 A26 Z38 U37 J2 L17 L12 C41 H84 H26 B77 F82 Q51 M50 K97 I38 M19 W62 Z89 O56 B44 I54 W64 Y58 B15 X8 D27 U40 U66 N63 F27 S62 K9 S3 R47 S78 R71 T12 E38 K1 P49 Q74 D036 C0 Y58 Z93 J84 E86 E3 U92
H08 M33 S90 R9 Q47 C58 O60 Z012 I45 J67 I30 P41 F22 I99 O022 M07 E61 E070 D0 O82 P68 S83 Z22 R65 N051 A56 V74 Y36 M88 M77 T41 A82 T75 B20 D37 Z26 Q37 S21 O61 E46 M07 U028 V19 N64 P60 P83 P34 S91 U11 A89 C82 B29 A41 Z98 U07 X80 M03 H63 E32 C18 T79 H61 L86 E82 G2
3 R59 P23 U44 N18 K7 D054 M25 B75 P4 P51 W57 E27 Q10 T033 G95 J49 O46 P9 A52 R088 M71 O43 W39 Z51 D32 Q9 Y78 O84 X43 Y38 T61 J068 B42 B47 H48 A19 W47 D39 Y46 D55 C17 P68 R21 J37 B92 X50 P066 H43 O74 O47 H63 P3 K80 W50 Q99 Y30 U84 H59 R58 Q70 Z76 H73 N59 B80 N87
F62 U55 V88 Q064 N99 Q68

Results for 600 items, averaged over 60 iterations:

Brute Force Strategy:
Average Time: 11.150 ms
Sorted Data:
A0 A025 A14 A18 A19 A26 A37 A41 A45 A50 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C5 C56 C58 C66 C82 C84 C92 C95 C99 D0
D036 D054 D16 D18 D2 D27 D29 D32 D32 D37 D39 D5 D55 D59 D65 D74 D81 D82 D87 D90 D99 E070 E08 E13 E14 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F
67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H43 H48 H5 H51 H53 H59 H63 H73 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 J99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
7 K7 K74 K77 K80 K80 K9 K94 K95 K97 K98 K99 L016 L060 L094 L12 L17 L17 L2 L26 L33 L53 L53 L61 L61 L66 L7 L70 L74 L77 L78 L82 L84 L86 L9 L9 L9 M0 M031 M041 M083 M13 M16 M19 M23 M25 M27 M31 M33 M34 M41 M43 M49 M50 M54 M54 M55 M63 M71 M77 M83 M84 M87 M88 N0 N0 N011
N02 N023 N051 N053 N076 N13 N15 N18 N22 N30 N42 N53 N59 N61 N61 N63 N64 N64 N65 N66 N67 N71 N74 N76 N87 N88 N97 N99 O022 O025 O032 O035 O081 O12 O15 O18 O19 O29 O41 O43 O46 O47 O47 O51 O56 O59 O6 O60 O61 O61 O71 O71 O73 O73 O74 O75 O76 O81 O84 O93 O96 P
037 P039 P066 P17 P18 P20 P23 P23 P28 P3 P33 P34 P38 P39 P4 P40 P40 P41 P42 P46 P48 P50 P51 P56 P56 P60 P63 P68 P68 P77 P82 P83 P83 P89 P9 P93 P96 Q0 Q064 Q10 Q12 Q14 Q16 Q17 Q2 Q20 Q26 Q30 Q34 Q37 Q47 Q51 Q68 Q70 Q70 Q74 Q89 Q9 Q9 Q92 Q99 R088 R14 R15 R
18 R21 R21 R26 R40 R47 R48 R55 R57 R57 R58 R59 R64 R65 R70 R71 R76 R83 R84 R86 R9 R94 R95 R97 R99 R99 S035 S084 S12 S2 S2 S21 S21 S25 S3 S31 S32 S45 S45 S45 S58 S60 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S91 S96 S98 T033 T11 T11 T12 T13 T21 T26 T28 T30 T31
T40 T41 T44 T45 T51 T6 T61 T75 T79 T89 T91 U0 U0 U028 U038 U049 U07 U11 U15 U16 U20 U23 U29 U34 U37 U40 U41 U44 U55 U57 U62 U62 U66 U84 U86 U86 U87 U91 U92 U94 U98 V093 V1 V11 V17 V19 V2 V26 V27 V3 V33 V34 V5 V54 V55 V55 V56 V59 V6 V64 V65 V74 V88 V92 W0 W1 W15
W19 W27 W33 W39 W47 W48 W49 W49 W50 W55 W57 W57 W6 W62 W64 W66 W73 W75 W77 X025 X3 X30 X36 X36 X39 X43 X43 X47 X50 X55 X59 X64 X67 X7 X8 X88 X89 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y30 Y36 Y38 Y38 Y4 Y40 Y42 Y43 Y46 Y5 Y56
Y57 Y58 Y58 Y64 Y78 Y80 Y80 Y85 Y87 Y9 Z012 Z034 Z071 Z1 Z10 Z15 Z22 Z26 Z3 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z51 Z61 Z65 Z7
Z76 Z89 Z93 Z93 Z97 Z98

Greedy Strategy:
Average Time: 3.822 ms
Sorted Data:
A0 A025 A14 A18 A19 A26 A37 A41 A41 A50 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C5 C56 C58 C66 C82 C84 C92 C95 C99 D0
D036 D054 D16 D18 D2 D27 D29 D32 D32 D37 D39 D5 D55 D59 D65 D74 D81 D82 D87 D90 D99 E070 E08 E13 E14 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F
67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H43 H48 H5 H51 H53 H59 H63 H73 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 J99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
7 K7 K74 K77 K80 K80 K9 K94 K95 K97 K98 K99 L016 L060 L094 L12 L17 L17 L2 L26 L33 L53 L53 L61 L61 L66 L7 L70 L74 L77 L78 L82 L84 L86 L9 L9 L9 M0 M031 M041 M083 M13 M16 M19 M23 M25 M27 M31 M33 M34 M41 M43 M49 M50 M54 M54 M55 M63 M71 M77 M83 M84 M87 M88 N0 N0 N011
N02 N023 N051 N053 N076 N13 N15 N18 N22 N30 N42 N53 N59 N61 N61 N63 N64 N64 N65 N66 N67 N71 N74 N76 N87 N88 N97 N99 O022 O025 O032 O035 O081 O12 O15 O18 O19 O29 O41 O43 O46 O47 O47 O51 O56 O59 O6 O60 O61 O61 O71 O71 O73 O73 O74 O75 O76 O81 O84 O93 O96 P
037 P039 P066 P17 P18 P20 P23 P23 P28 P3 P33 P34 P38 P39 P4 P40 P40 P41 P42 P46 P48 P50 P51 P56 P56 P60 P63 P68 P68 P77 P82 P83 P83 P89 P9 P93 P96 Q0 Q064 Q10 Q12 Q14 Q16 Q17 Q2 Q20 Q26 Q30 Q34 Q37 Q47 Q51 Q68 Q70 Q70 Q74 Q89 Q9 Q9 Q92 Q99 R088 R14 R15 R
18 R21 R21 R26 R40 R47 R48 R55 R57 R57 R58 R59 R64 R65 R70 R71 R76 R83 R84 R86 R9 R94 R95 R97 R99 R99 S035 S084 S12 S2 S2 S21 S21 S25 S3 S31 S32 S45 S45 S45 S58 S60 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S91 S96 S98 T033 T11 T11 T12 T13 T21 T26 T28 T30 T31
T40 T41 T44 T45 T51 T6 T61 T75 T79 T89 T91 U0 U0 U028 U038 U049 U07 U11 U15 U16 U20 U23 U29 U34 U37 U40 U41 U44 U55 U57 U62 U62 U66 U84 U86 U86 U87 U91 U92 U94 U98 V093 V1 V11 V17 V19 V2 V26 V27 V3 V33 V34 V5 V54 V55 V55 V56 V59 V6 V64 V65 V74 V88 V92 W0 W1 W15
W19 W27 W33 W39 W47 W48 W49 W49 W50 W55 W57 W57 W6 W62 W64 W66 W73 W75 W77 X025 X3 X30 X36 X36 X39 X43 X43 X47 X50 X55 X59 X64 X67 X7 X8 X88 X89 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y30 Y36 Y38 Y38 Y4 Y40 Y42 Y43 Y46 Y5 Y56
Y57 Y58 Y58 Y64 Y78 Y80 Y80 Y85 Y87 Y9 Z012 Z034 Z071 Z1 Z10 Z15 Z22 Z26 Z3 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z51 Z58 Z60 Z61 Z65 Z65 Z68 Z7 Z72 Z76 Z89 Z89 Z93 Z93 Z97 Z98

Bubble Strategy:
Average Time: 30.828 ms
Sorted Data:
A0 A025 A14 A18 A19 A26 A37 A41 A41 A50 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C5 C56 C58 C66 C82 C84 C92 C95 C99 D0
D036 D054 D16 D18 D2 D27 D29 D32 D32 D37 D39 D5 D55 D59 D65 D74 D81 D82 D87 D90 D99 E070 E08 E13 E14 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F018 F076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F
67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G3 G35 G38 G42 G50 G56 G63 G67 G87 G88 G9 G95 G98 H088 H19 H26 H27 H32 H34 H36 H43 H48 H5 H51 H53 H59 H63 H73 H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 J99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
7 K7 K74 K77 K80 K80 K9 K94 K95 K97 K98 K99 L016 L060 L094 L12 L17 L17 L2 L26 L33 L53 L53 L61 L61 L66 L7 L70 L74 L77 L78 L82 L84 L86 L9 L9 L9 M0 M031 M041 M083 M13 M16 M19 M23 M25 M27 M31 M33 M34 M41 M43 M49 M50 M54 M54 M55 M63 M71 M77 M83 M84 M87 M88 N0 N0 N011
N02 N023 N051 N053 N076 N13 N15 N18 N22 N30 N42 N53 N59 N61 N61 N63 N64 N64 N65 N66 N67 N71 N74 N76 N87 N88 N97 N99 O022 O025 O032 O035 O081 O12 O15 O18 O19 O29 O41 O43 O46 O47 O47 O51 O56 O59 O6 O60 O61 O61 O71 O71 O73 O73 O74 O75 O76 O81 O84 O93 O96 P
037 P039 P066 P17 P18 P20 P23 P23 P28 P3 P33 P34 P38 P39 P4 P40 P40 P41 P42 P46 P48 P50 P51 P56 P56 P60 P63 P68 P68 P77 P82 P83 P83 P89 P9 P93 P96 Q0 Q064 Q10 Q12 Q14 Q16 Q17 Q2 Q20 Q26 Q30 Q34 Q37 Q47 Q51 Q68 Q70 Q70 Q74 Q89 Q9 Q9 Q92 Q99 R088 R14 R15 R
18 R21 R21 R26 R40 R47 R48 R55 R57 R57 R58 R59 R64 R65 R70 R71 R76 R83 R84 R86 R9 R94 R95 R97 R99 R99 S035 S084 S2 S2 S21 S21 S25 S3 S31 S32 S45 S45 S45 S58 S60 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S91 S96 S98 T033 T11 T11 T12 T13 T21 T26 T28 T30 T31
T40 T41 T44 T45 T51 T6 T61 T75 T79 T89 T91 U0 U0 U028 U038 U049 U07 U11 U15 U16 U20 U23 U29 U34 U37 U40 U41 U44 U55 U57 U62 U62 U66 U84 U86 U86 U87 U91 U92 U94 U98 V093 V1 V11 V17 V19 V2 V26 V27 V3 V33 V34 V5 V54 V55 V55 V56 V59 V6 V64 V65 V74 V88 V92 W0 W1 W15
W19 W27 W33 W39 W47 W48 W49 W49 W50 W55 W57 W57 W6 W62 W64 W66 W73 W75 W77 X025 X3 X30 X36 X36 X39 X43 X43 X47 X50 X55 X59 X64 X67 X7 X8 X88 X89 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y30 Y36 Y38 Y38 Y4 Y40 Y42 Y43 Y46 Y5 Y56
Y57 Y58 Y58 Y64 Y78 Y80 Y80 Y85 Y87 Y9 Z012 Z034 Z071 Z1 Z10 Z15 Z22 Z26 Z3 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z51 Z58 Z60 Z61 Z65 Z65 Z68 Z7 Z72 Z76 Z89 Z89 Z93 Z93 Z97 Z98
```



```
Enter problem size: 700
Enter number of iterations: 70
Original Data:
P56 H34 Q0 G87 H95 U16 M31 E70 A025 Y57 L33 N023 L17 F096 D65 X47 F6 I1 R84 C28 V92 S084 C56 X88 G0 G26 B33 W93 K47 F97 N76 Q70 D16 U049 X64 W66 F85 N053 F30 O032 Z41 V56 S69 R57 T31 K99 J58 P39 A40 E14 P63 G35 G15 X9 R15 P83 N61 R83 V11 Y80 S25 T6 M34 W49 C84 Y
38 S45 Y27 Y19 C32 Q70 P20 E46 V33 I98 K77 E13 F5 F50 M55 Z43 W42 I7 M13 K95 K17 A41 S45 V2 W55 S70 B48 E44 N67 Z1 K41 Y20 C95 X89 F98 X019 T21 L2 S45 G22 Y80 P63 S91 F81 A9 D5 P037 O61 O19 E41 F45 X88 Z44 B27 C16 O61 E62 J49 U88 V6 P40 V54 A37 E30 O41 Y42 G63 P
18 O73 E08 Y56 L70 F73 P83 B16 N88 P96 L74 J50 V55 R86 V5 I39 P50 Y4 A7 M41 Y11 E61 H088 W77 Q2 N74 Q38 R18 Q12 P33 M16 X25 U038 J43 T32 L82 O76 O47 V64 A64 O73 W82 U0 X30 W19 H94 M23 S69 N64 C99 B59 X77 C43 O035 K1 S72 F55 Z93 K45 V34 A08 T30 X59 D18 K80 N15 L8
4 Y40 J93 Y85 H32 F076 I46 X74 J1 S21 W33 N99 K12 K7 U34 F53 N0 U29 X36 X17 Z10 R64 Z97 B070 M84 N02 M083 G50 Q20 O881 O51 K74 E55 T89 L9 Y19 H81 N65 K58 O71 A87 U57 G016 Z43 Q34 R57 C92 D59 D90 I78 U62 T11 E18 I029 O81 J1 W48 A14 Y87 Y43 Z7 P38 V3 S32 C061 M27
P56 S98 A50 J26 L66 F46 V55 G21 U41 B79 F52 A89 A10 O59 V093 L9 Z65 Y9 L61 N011 T44 R97 K52 D99 C66 P82 W15 S60 R14 T45 E87 S035 J093 W75 H36 O74 I56 Z071 C33 O025 B22 Z15 C066 N0 F91 W6 U016 Q14 I15 J59 T13 V65 D81 W49 V17 X3 H27 R99 C5 B113 L53 D87 V1 G0 Y5 L53
N73 G88 B94 U20 S58 B34 M031 B028 O29 R99 X13 T26 L7 H5 C87 S2 M83 P17 X90 O12 H59 G42 M8 G38 N076 K25 E56 U91 F67 D29 X0 O15 T28 O93 G95 B82 G98 X17 P40 E41 V64 A0 N71 F018 E60 T40 C0 U86 K94 E91 P23 Z61 S62 H81 K21 L9 J018 U86 G56 G25 G67 E87 K32 J04 D32 Q0 Z
34 J11 E55 N61 P42 E50 V27 Q26 G3 X95 I79 E60 P28 J84 S62 R55 D2 Z41 J90 A26 Z38 U37 J2 L17 L12 C41 H84 H26 B77 F82 Q51 M50 K97 I38 M19 W62 Z89 O56 B44 I54 W64 Y58 B15 X8 D27 U40 U66 N63 F27 S62 K9 S3 R47 S78 R71 T12 E38 K1 P49 Q74 D036 C0 Y58 Z93 J84 E86 E3 U92
H08 M33 S90 R9 Q47 C58 O60 Z012 I45 J67 I30 P41 F22 I99 O022 M07 E61 E070 D0 O82 P68 S83 Z22 R65 N051 A56 V74 Y36 M88 M77 T41 A82 T75 B20 D37 Z26 Q37 S21 O61 E46 M07 U028 V19 N64 P60 P83 P34 S91 U11 A89 C82 B29 A41 Z98 U07 X80 M03 H63 E32 C18 T79 H61 L86 E82 G2
3 R59 P23 U44 N18 K7 D054 M25 B75 P4 P51 W57 E27 Q10 T033 G95 J49 O46 P9 A52 R088 M71 O43 W39 Z51 D32 Q9 Y78 O84 X43 Y38 T61 J068 B42 B47 H48 A19 W47 D39 Y46 D55 C17 P68 R21 J37 B92 X50 P066 H43 O74 O47 H63 P3 K80 W50 Q99 Y30 U84 H59 R58 Q70 Z76 H73 N59 B80 N87
F62 U55 V88 Q064 N99 Q68 Q92 L61 R26 Z68 J84 P039 X15 I88 O6 G93 V59 L060 I55 E18 X67 D48 Z89 U0 Z72 O71 S31 R95 K98 R70 U87 596 V26 O18 J38 B23 R94 Z65 M54 W57 Z034 H48 O75 W27 K69 J53 I30 L26 E065 F00 R21 A019 A51 M041 D061 I68 G65 P93 N53 P46 U23 U15 H023 G29
X43 P77 Q89 N66 J32 W1 J14 M43 W0 A45 X36 X39 M54 N22 S12 N30 H72 H39 L78 Q16 Q17 T51 F037 Z58 O96 T91 C43 361 T11 N13 Z60 X55 U94 G81 L094 S2 D867

Results for 700 items, averaged over 70 iterations:

Brute Force Strategy:
Average Time: 15.148 ms
Sorted Data:
A0 A019 A025 A14 A18 A19 A26 A37 A41 A41 A45 A50 A51 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B23 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C43 C5 C56 C58 C66 C8
2 C84 C92 C95 C99 D0 D036 D054 D061 D087 D16 D18 D2 D27 D29 D32 D32 D37 D39 D48 D5 D55 D59 D65 D74 D81 D82 D87 D9 D90 D99 E065 E070 E08 E13 E14 E18 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F00 F018 F037 F
076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G29 G3 G35 G38 G42 G50 G56 G63 G65 G67 G81 G87 G88 G9 G93 G95 G95 G98 H023 H088 H19 H26 H27 H32 H34 H36 H39 H43 H48 H5 H51 H53 H59 H63 H72 H73
H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 I99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
7 K7 K74 K77 K80 K80 K9 K94 K95 K97 K98 K99 L016 L060 L094 L12 L17 L17 L2 L26 L33 L53 L53 L61 L61 L66 L7 L70 L74 L77 L78 L82 L84 L86 L9 L9 L9 M0 M031 M041 M083 M13 M16 M19 M23 M25 M27 M31 M33 M34 M41 M43 M49 M50 M54 M54 M55 M63 M71 M77 M83 M84 M87 M88 N0 N0 N011
N02 N023 N051 N053 N076 N13 N15 N18 N22 N30 N42 N53 N59 N61 N61 N63 N64 N64 N65 N66 N67 N71 N74 N76 N87 N88 N97 N99 O022 O025 O032 O035 O081 O12 O15 O18 O19 O29 O41 O43 O46 O47 O47 O51 O56 O59 O6 O60 O61 O61 O71 O71 O73 O73 O74 O75 O76 O81 O84 O93 O96 P
037 P039 P066 P17 P18 P20 P23 P23 P28 P3 P33 P34 P38 P39 P4 P40 P40 P41 P42 P46 P48 P50 P51 P56 P56 P60 P63 P68 P68 P77 P82 P83 P83 P89 P9 P93 P96 Q0 Q064 Q10 Q12 Q14 Q16 Q17 Q2 Q20 Q26 Q30 Q34 Q37 Q47 Q51 Q68 Q70 Q70 Q74 Q89 Q9 Q9 Q92 Q99 R088 R14 R15 R
18 R21 R21 R26 R40 R47 R48 R55 R57 R57 R58 R59 R64 R65 R70 R71 R76 R83 R84 R86 R9 R94 R95 R97 R99 R99 S035 S084 S12 S2 S2 S21 S21 S25 S3 S31 S32 S45 S45 S45 S58 S60 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S91 S96 S98 T033 T11 T11 T12 T13 T21 T26 T28 T30 T31
T40 T41 T44 T45 T51 T6 T61 T75 T79 T89 T91 U0 U0 U028 U038 U049 U07 U11 U15 U16 U20 U23 U29 U34 U37 U40 U41 U44 U55 U57 U62 U62 U66 U84 U86 U86 U87 U91 U92 U94 U98 V093 V1 V11 V17 V19 V2 V26 V27 V3 V33 V34 V5 V54 V55 V55 V56 V59 V6 V64 V65 V74 V88 V92 W0 W1 W15
W19 W27 W33 W39 W47 W48 W49 W49 W50 W55 W57 W57 W6 W62 W64 W66 W73 W75 W77 X025 X3 X30 X36 X36 X39 X43 X43 X47 X50 X55 X59 X64 X67 X7 X8 X88 X89 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y30 Y36 Y38 Y38 Y4 Y40 Y42 Y43 Y46 Y5 Y56
Y57 Y58 Y58 Y64 Y78 Y80 Y80 Y85 Y87 Y9 Z012 Z034 Z071 Z1 Z10 Z15 Z22 Z26 Z3 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z51 Z58 Z60 Z61 Z65 Z65 Z68 Z7 Z72 Z76 Z89 Z89 Z93 Z93 Z97 Z98

Greedy Strategy:
Average Time: 4.952 ms
Sorted Data:
A0 A019 A025 A14 A18 A19 A26 A37 A41 A41 A45 A50 A51 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B23 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C43 C5 C56 C58 C66 C8
2 C84 C92 C95 C99 D0 D036 D054 D061 D087 D16 D18 D2 D27 D29 D32 D32 D37 D39 D48 D5 D55 D59 D65 D74 D81 D82 D87 D9 D90 D99 E065 E070 E08 E13 E14 E18 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F00 F018 F037 F
076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G29 G3 G35 G38 G42 G50 G56 G63 G65 G67 G81 G87 G88 G9 G93 G95 G95 G98 H023 H088 H19 H26 H27 H32 H34 H36 H39 H43 H48 H5 H51 H53 H59 H63 H72 H73
H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 I99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
7 K7 K74 K77 K80 K80 K9 K94 K95 K97 K98 K99 L016 L060 L094 L12 L17 L17 L2 L26 L33 L53 L53 L61 L61 L66 L7 L70 L74 L77 L78 L82 L84 L86 L9 L9 L9 M0 M031 M041 M083 M13 M16 M19 M23 M25 M27 M31 M33 M34 M41 M43 M49 M50 M54 M54 M55 M63 M71 M77 M83 M84 M87 M88 N0 N0 N011
N02 N023 N051 N053 N076 N13 N15 N18 N22 N30 N42 N53 N59 N61 N61 N63 N64 N64 N65 N66 N67 N71 N74 N76 N87 N88 N97 N99 O022 O025 O032 O035 O081 O12 O15 O18 O19 O29 O41 O43 O46 O47 O47 O51 O56 O59 O6 O60 O61 O61 O71 O71 O73 O73 O74 O75 O76 O81 O84 O93 O96 P
037 P039 P066 P17 P18 P20 P23 P23 P28 P3 P33 P34 P38 P39 P4 P40 P40 P41 P42 P46 P48 P50 P51 P56 P56 P60 P63 P68 P68 P77 P82 P83 P83 P89 P9 P93 P96 Q0 Q064 Q10 Q12 Q14 Q16 Q17 Q2 Q20 Q26 Q30 Q34 Q37 Q47 Q51 Q68 Q70 Q70 Q74 Q89 Q9 Q9 Q92 Q99 R088 R14 R15 R
18 R21 R21 R26 R40 R47 R48 R55 R57 R57 R58 R59 R64 R65 R70 R71 R76 R83 R84 R86 R9 R94 R95 R97 R99 R99 S035 S084 S12 S2 S2 S21 S21 S25 S3 S31 S32 S45 S45 S45 S58 S60 S62 S62 S69 S69 S70 S72 S78 S83 S90 S91 S91 S96 S98 T033 T11 T11 T12 T13 T21 T26 T28 T30 T31
T40 T41 T44 T45 T51 T6 T61 T75 T79 T89 T91 U0 U0 U028 U038 U049 U07 U11 U15 U16 U20 U23 U29 U34 U37 U40 U41 U44 U55 U57 U62 U62 U66 U84 U86 U86 U87 U91 U92 U94 U98 V093 V1 V11 V17 V19 V2 V26 V27 V3 V33 V34 V5 V54 V55 V55 V56 V59 V6 V64 V65 V74 V88 V92 W0 W1 W15
W19 W27 W33 W39 W47 W48 W49 W49 W50 W55 W57 W57 W6 W62 W64 W66 W73 W75 W77 X025 X3 X30 X36 X36 X39 X43 X43 X47 X50 X55 X59 X64 X67 X7 X8 X88 X89 X89 X9 X90 X95 Y11 Y19 Y19 Y20 Y27 Y30 Y36 Y38 Y38 Y4 Y40 Y42 Y43 Y46 Y5 Y56
Y57 Y58 Y58 Y64 Y78 Y80 Y80 Y85 Y87 Y9 Z012 Z034 Z071 Z1 Z10 Z15 Z22 Z26 Z3 Z34 Z38 Z41 Z41 Z43 Z43 Z44 Z51 Z58 Z60 Z61 Z65 Z65 Z68 Z7 Z72 Z76 Z89 Z89 Z93 Z93 Z97 Z98

Bubble Strategy:
Average Time: 14.276 ms
Sorted Data:
A0 A019 A025 A14 A18 A19 A26 A37 A41 A41 A45 A50 A51 A52 A56 A64 A7 A82 A87 A89 A89 A9 A98 B028 B070 B13 B15 B16 B20 B22 B23 B27 B29 B30 B33 B34 B42 B44 B47 B59 B75 B77 B79 B80 B82 B92 B94 C0 C0 C061 C066 C07 C16 C17 C18 C28 C32 C33 C41 C43 C43 C5 C56 C58 C66 C8
2 C84 C92 C95 C99 D0 D036 D054 D061 D087 D16 D18 D2 D27 D29 D32 D32 D37 D39 D48 D5 D55 D59 D65 D74 D81 D82 D87 D9 D90 D99 E065 E070 E08 E13 E14 E18 E18 E27 E3 E32 E38 E41 E41 E44 E46 E46 E50 E55 E55 E56 E60 E60 E61 E61 E62 E70 E82 E86 E87 E87 E91 F00 F018 F037 F
076 F096 F22 F27 F30 F45 F46 F5 F50 F52 F53 F55 F6 F62 F67 F73 F81 F82 F85 F91 F97 F98 G0 G016 G15 G21 G22 G23 G25 G26 G29 G3 G35 G38 G42 G50 G56 G63 G65 G67 G81 G87 G88 G9 G93 G95 G95 G98 H023 H088 H19 H26 H27 H32 H34 H36 H39 H43 H48 H5 H51 H53 H59 H63 H72 H73
H81 H81 H84 H94 H95 H98 I029 I1 I15 I30 I38 I39 I45 I46 I54 I55 I56 I68 I7 I78 I79 I88 I98 I99 J018 J04 J068 J093 J1 J1 J11 J14 J2 J26 J32 J32 J37 J38 J43 J49 J49 J50 J53 J58 J59 J61 J67 J84 J84 J84 J90 J93 K1 K1 K12 K17 K21 K25 K32 K41 K45 K47 K52 K58 K69 K
```


Enter problem size: 800
Enter number of iterations: 80
Original Data:
Results for 800 items, averaged over 80 iterations:
Brute Force Strategy:
Average Time: 19.500 ms
Sorted Data:
Greedy Strategy:
Average Time: 6.714 ms
Sorted Data:


Enter problem size: 900
Enter number of iterations: 90
Original Data:
Results for 900 items, averaged over 90 iterations:
Brute Force Strategy:
Average Time: 23.981 ms
Sorted Data:
Greedy Strategy:
Average Time: 8.172 ms
Sorted Data:



Tabel Hasil Perbandingan Pada Proses Iteration

No	Problem Size (n)	Iteration	Brute Force	Greedy	Bubble	Insertion	Merge
1	100	10	0.800	0.817	0.825	0.101	0.104
2	200	20	1.194	0.786	1.092	0.724	0.409
3	300	30	2.887	0.906	2.749	1.367	0.580
4	400	40	5.005	1.657	4.748	2.631	0.931
5	500	50	7.575	2.608	7.712	3.848	1.015
6	600	60	11.150	3.822	10.828	5.709	1.243
7	700	70	15.148	4.952	14.276	7.207	1.538
8	800	80	19.500	6.714	19.397	9.805	1.859
9	900	90	23.981	8.172	23.602	12.293	2.111
0	1000	100	31.175	10.520	31.148	16.009	2.423

BAB IV
ANALISIS HASIL PENGUJIAN

Dari hasil pengujian yang dilakukan terhadap implementasi lima algoritma pengurutan yang berbeda (Brute Force, Greedy/Selection Sort, Bubble Sort, Insertion Sort, dan Merge Sort), dapat dianalisis beberapa aspek penting sebagai berikut:

1. Perbandingan Kompleksitas Waktu Algoritma

a) Brute Force Strategy memiliki kompleksitas waktu $O(n^2)$ karena pada setiap iterasi harus mencari elemen minimum dari seluruh data yang tersisa. Hal ini membuat algoritma ini kurang efisien untuk dataset besar.

b) Greedy Strategy (Selection Sort) juga memiliki kompleksitas $O(n^2)$, namun dalam implementasinya lebih efisien dari Brute Force karena melakukan pertukaran langsung tanpa membuat array baru.

c) Bubble Sort dengan kompleksitas $O(n^2)$ menunjukkan performa yang mirip dengan Selection Sort, namun cenderung lebih lambat karena melakukan lebih banyak operasi pertukaran.

d) Insertion Sort dengan kompleksitas $O(n^2)$ menunjukkan performa yang lebih baik untuk dataset yang hampir terurut, karena dapat mengurangi jumlah pergeseran yang diperlukan.

e) Merge Sort dengan kompleksitas $O(n \log n)$ menunjukkan performa terbaik untuk dataset besar, meskipun membutuhkan ruang tambahan untuk proses penggabungan.

2. Analisis Penggunaan Memori

a) Brute Force Strategy memerlukan memori tambahan untuk menyimpan array hasil pengurutan, yang membuat penggunaan memorinya kurang efisien.

b) Greedy, Bubble, dan Insertion Sort melakukan pengurutan secara in-place, sehingga hanya membutuhkan memori tambahan $O(1)$.

c) Merge Sort membutuhkan memori tambahan $O(n)$ untuk proses penggabungan (merge), yang menjadi trade-off untuk kecepatan prosesnya.

3. Karakteristik Performa pada Dataset yang Berbeda

a) Untuk dataset kecil ($n < 100$):

- Semua algoritma menunjukkan performa yang relatif setara
- Insertion Sort dan Bubble Sort dapat lebih efisien jika data sudah hampir terurut

b) Untuk dataset menengah ($100 < n < 1000$):

- Merge Sort mulai menunjukkan keunggulannya
- Brute Force dan Bubble Sort mulai menunjukkan penurunan performa yang signifikan

c) Untuk dataset besar ($n > 1000$):

- Merge Sort secara konsisten menunjukkan performa terbaik
- Algoritma dengan kompleksitas $O(n^2)$ menunjukkan peningkatan waktu eksekusi yang sangat signifikan

4. Stabilitas Algoritma

a. Merge Sort merupakan algoritma yang stabil, mempertahankan urutan relatif elemen-elemen yang memiliki nilai sama
b. Bubble Sort dan Insertion Sort juga merupakan algoritma yang stabil
c. Selection Sort (Greedy) tidak menjamin stabilitas urutan untuk elemen-elemen dengan nilai yang sama

5. Pengaruh Jumlah Iterasi

a. Semakin banyak iterasi yang dilakukan, semakin akurat rata-rata waktu eksekusi yang didapatkan
b. Fluktuasi waktu eksekusi antar iterasi relatif kecil untuk dataset yang sama
c. Perbedaan performa antar algoritma semakin terlihat jelas dengan bertambahnya jumlah iterasi

Kesimpulan dari analisis ini menunjukkan bahwa:

1. Merge Sort merupakan pilihan terbaik untuk dataset besar karena kompleksitas waktunya yang lebih efisien
2. Insertion Sort dapat menjadi pilihan yang baik untuk dataset kecil atau data yang hampir terurut
3. Algoritma dengan kompleksitas $O(n^2)$ seperti Brute Force, Selection Sort, dan Bubble Sort sebaiknya dihindari untuk dataset besar

a) 4. Pemilihan algoritma pengurutan harus mempertimbangkan trade-off antara kecepatan eksekusi, penggunaan memori, dan karakteristik dataset yang akan diurutkan.

BAB V
KESIMPULAN

Berdasarkan hasil implementasi dan analisis pengujian dari kelima algoritma pengurutan (Brute Force, Greedy/Selection Sort, Bubble Sort, Insertion Sort, dan Merge Sort) dalam mengurutkan kombinasi string yang terdiri dari satu huruf dan dua digit angka, dapat ditarik beberapa kesimpulan penting:

1. Efisiensi Algoritma

a. Merge Sort menunjukkan performa terbaik untuk dataset besar dengan kompleksitas waktu O(n log n), meskipun membutuhkan memori tambahan

b. Algoritma dengan kompleksitas $O(n^2)$ seperti Brute Force, Selection Sort, dan Bubble Sort kurang efisien untuk dataset besar

c. Insertion Sort menunjukkan performa yang baik untuk dataset kecil atau data yang hampir terurut

2. Trade-off Implementasi
    a. Terdapat trade-off antara kecepatan eksekusi dan penggunaan memori, dimana algoritma yang lebih cepat seperti Merge Sort membutuhkan memori tambahan
    b. Algoritma in-place seperti Selection Sort dan Bubble Sort menggunakan memori minimal tetapi memiliki waktu eksekusi yang lebih lama

3. Karakteristik Dataset
    a. Ukuran dataset sangat mempengaruhi performa relatif dari setiap algoritma
    b. Kondisi awal dataset (sudah terurut sebagian, acak, atau terbalik) mempengaruhi efisiensi algoritma tertentu seperti Insertion Sort

4. Rekomendasi Penggunaan
    a. Untuk aplikasi dengan dataset besar: Gunakan Merge Sort
    b. Untuk dataset kecil atau memori terbatas: Gunakan Insertion Sort
    c. Untuk pembelajaran atau dataset sangat kecil: Bubble Sort atau Selection Sort dapat digunakan karena implementasinya yang sederhana

5. Pertimbangan Implementasi
    a. Pemilihan algoritma pengurutan harus mempertimbangkan karakteristik data, ketersediaan memori, dan kebutuhan performa
    b. Stabilitas pengurutan perlu dipertimbangkan jika urutan relatif elemen dengan nilai sama harus dipertahankan

Hasil pengujian dan analisis ini menunjukkan bahwa tidak ada algoritma pengurutan yang "terbaik" untuk semua situasi. Pemilihan algoritma yang tepat harus mempertimbangkan berbagai faktor seperti ukuran data, karakteristik data, ketersediaan memori, dan kebutuhan performa spesifik dari aplikasi yang dikembangkan.

REFRENSI

Hendra Saptadi, Arief . 2013, *Analisis Algoritma Insertion sort, Merge sort dan Implementasinya dalam Bahasa Pemograman C++*. Akademi Teknik Telekomunikasi Shandy Putra Purwokerto

Firmansyah, T., & Maulana, H. (2020). *Analisis Perbandingan Algoritma Bubble Sort, Merge Sort dan Quick Sort dalam Proses Pengurutan Kombinasi Angka dan Huruf*. Jurnal Teknik Informatika, 13(2), 169-176.

Pranata, D., & Yulianto, S. (2021). *Implementasi dan Analisis Perbandingan Algoritma Bubble Sort, Selection Sort, dan Insertion Sort pada Pengurutan Data*. InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan, 6(1), 110-115.

Hendra Saptadi, Arief . 2013, *Analisis Algoritma Insertion sort, Merge sort dan Implementasinya dalam Bahasa Pemograman C++*. Akademi Teknik Telekomunikasi Shandy Putra Purwokerto