**DSA Homework 1 Report**

Haocong Wang

[mw814@scarletmail.rutgers.edu](mailto:mw814@scarletmail.rutgers.edu)

**Question 1**

|  |  |  |
| --- | --- | --- |
| Size | O(N^3)(ms) | O(N^2lgN)(ms) |
| 8 | 0.004 | 0.005 |
| 32 | 0.049 | 0.071 |
| 128 | 3.274 | 0.99 |
| 512 | 193.615 | 20.405 |
| 1024 | 1548.38 | 78.406 |
| 4096 | 92434.9 | 1419.19 |
| 4192 | 100535 | 1500.41 |
| 8192 | 742342 | 5962.08 |

In question 1, I took advantage of the cftool in MATLAB. Detailed analysis will be shown in question 3.

**1. Naive 3-sum**

图片包含 文字

描述已自动生成

From this figure and the data, I believe the naïve 3-sum algorithm is a O(N^3) algorithm.

**2. Sophisticated 3-sum**

图片包含 文字

描述已自动生成

From this figure and the data, I believe the sophisticated 3-sum is a O(N^2lgN) algorithm.

**Question 2**

|  |  |  |  |
| --- | --- | --- | --- |
| Size | Quick find (ms) | Quick union (ms) | Weighted quick union (ms) |
| 8 | 0.011 | 0.001 | 0.001 |
| 32 | 0.014 | 0.002 | 0.001 |
| 128 | 0.017 | 0.004 | 0.003 |
| 512 | 0.409 | 0.073 | 0.023 |
| 1024 | 0.425 | 0.312 | 0.031 |
| 4096 | 25.273 | 8.72 | 0.165 |
| 8192 | 131.815 | 6.055 | 0.378 |

In question 2, I also used the cftool to finish the curve fitting of the data. Detailed analysis will be shown in question 3.

**1. Quick find**

图片包含 文字

描述已自动生成

In the curve fitting, I tried O(N) and O(N^2) and they both worked well, but the SSE of O(N^2) is smaller, so I think quick find is a O(N^2) algorithm in the worst case.

**2. Quick union**

图片包含 文字

描述已自动生成

It is obvious that quick union is at least a O(MN) algorithm and it could be O(N^2) in the worst case.

**3. Weighted quick union**

图片包含 文字

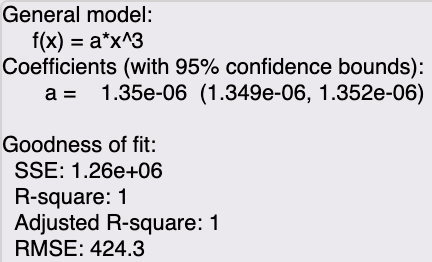
描述已自动生成

From this figure and the data, we call tell that weighted quick union is a O(M\*lgN) algorithm. In the worst case, it could be O(N\*lgN).

**Question 3**

**1. Naive 3-sum**

图片包含 文字

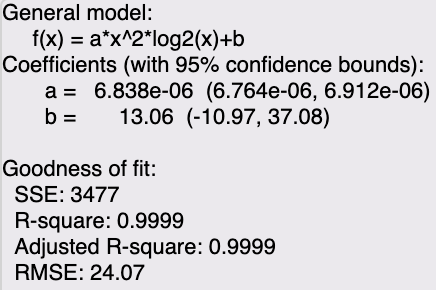
描述已自动生成

Suppose that f(N) = 1.35\*10^(-6)\*N^3.

Assume that g(N) = N^3 and c = 1. To get f(N) < c\*g(N) (N>Nc), we can set Nc = 1.

**2. Sophisticated 3-sum**

图片包含 文字

描述已自动生成

Suppose that f(N) = 6.838\*10-6\*N2\*log(N)+13.06.

Assume that g(N) = N2\*log(N) and c =7. To get f(N) < c\*g(N) (N>Nc), we can set Nc = 10.

**3. Quick find**

图片包含 文字

描述已自动生成图片包含 文字

描述已自动生成

Suppose that f(N) = 2.401\*10-6\*N^2-0.0036\*N+0.6929.

Assume that g(N) = N^2 and c = 3. We can get Nc = 1.

**4. Quick union**

图片包含 文字

描述已自动生成图片包含 文字

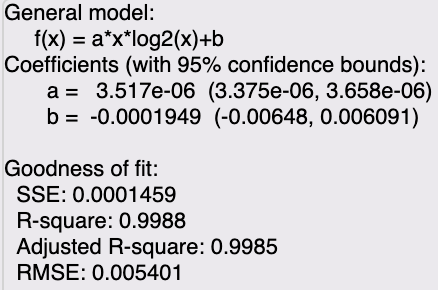
描述已自动生成

Suppose that f(N) = -2.986\*10-7\*N^2+0.003311\*N-0.8164.

Assume that g(N) = N^2 and c = -1. We can get Nc = 1.

**5. Weighted quick union**

图片包含 文字

描述已自动生成

Suppose that f(N) = 3.517\*10-6\*N\*log(N)-0.0001949.

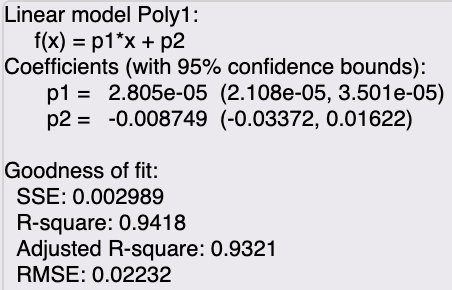
Assume that g(N) = N\*log(N) and c = 4. We can get Nc = 2.

**Question 4**

The data I used in this question is as same as question 1.

|  |  |
| --- | --- |
| Size | Time(ms) |
| 8 | 0.001 |
| 32 | 0.001 |
| 128 | 0.002 |
| 512 | 0.008 |
| 1024 | 0.011 |
| 4096 | 0.042 |
| 4192 | 0.086 |
| 8192 | 0.108 |

图片包含 文字, 地图, 室内

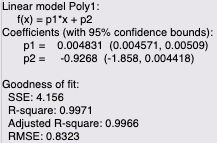
描述已自动生成

It is obvious that this is a linear algorithm.

**Question 5**

|  |  |
| --- | --- |
| Size | Time(ms) |
| 8 | 0.002 |
| 32 | 0.006 |
| 128 | 0.073 |
| 512 | 0.094 |
| 1024 | 2.951 |
| 4096 | 18.324 |
| 4192 | 18.766 |
| 8192 | 39.36 |

图片包含 文字

描述已自动生成

Apparently, this is O(N) algorithm, which is faster than the two previous algorithms.