DSA Homework 1 Report

Haocong Wang

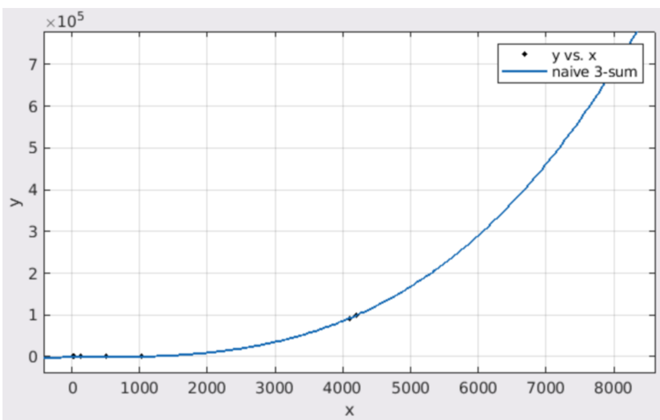
mw814@scarletmail.rutgers.edu

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

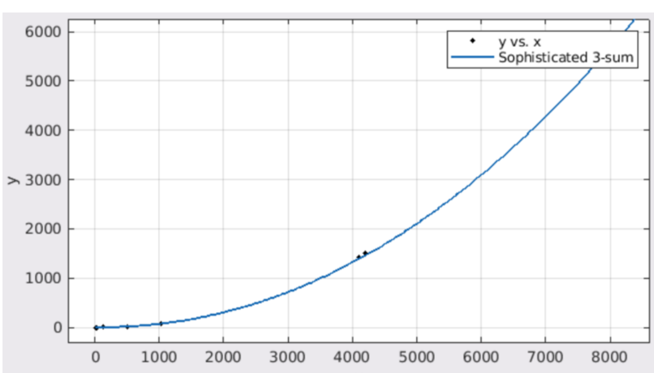
In question 1, I took advantage of the cftool in MATLAB. Detailed analysis will be shown in question 3.

1. Naive 3-sum



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

2. Sophisticated 3-sum



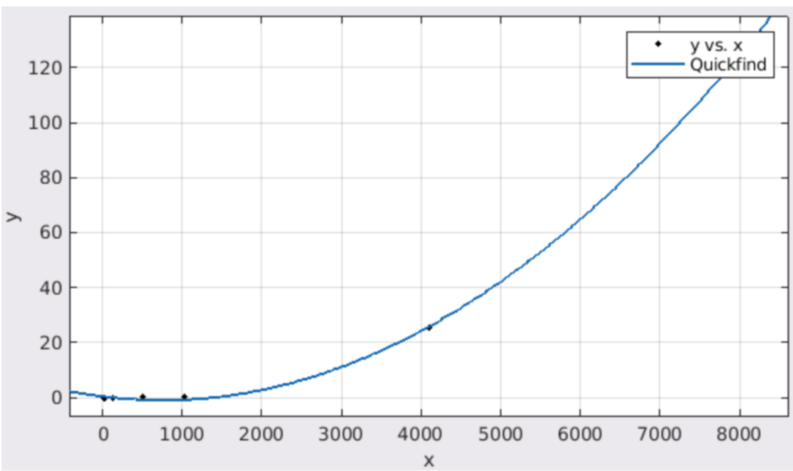
From this figure and the data, I believe the sophisticated 3-sum is a O(N^2lgN) algorithm.

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

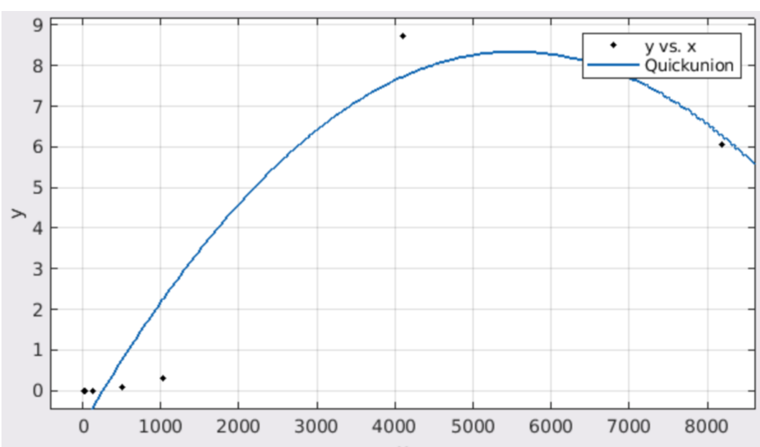
In question 2, I also used the cftool to finish the curve fitting of the data. Detailed analysis will be shown in question 3.

1. Quick find



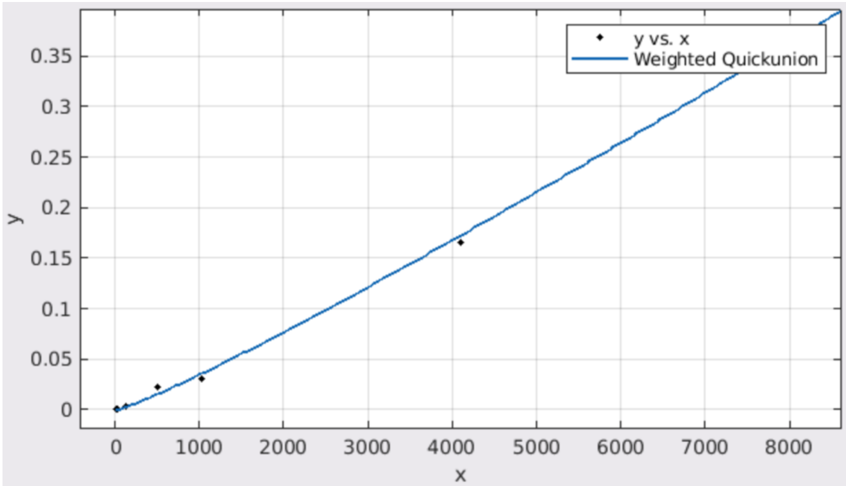
In the curve fitting, I tried O(N) and O(N^2) and they both worked well, but the SSE of O(N^2) is smaller, so I think quick find is a O(N^2) algorithm in the worst case.

2. Quick union



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

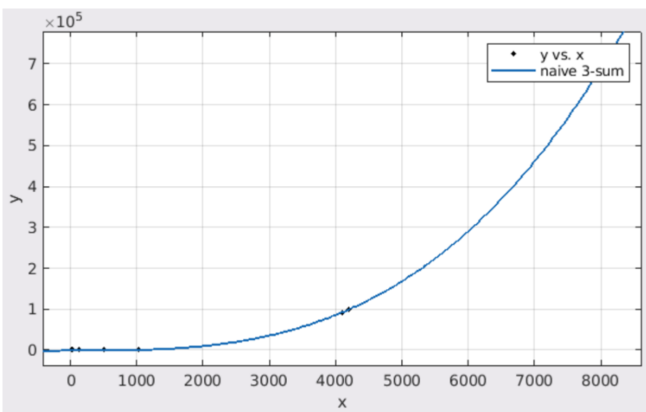
3. Weighted quick union



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



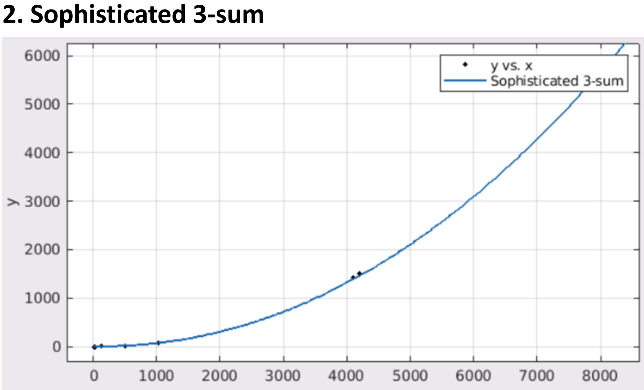
General model:
f(x) = a*x^3
Coefficients (with 95% confidence bounds):
a = 1.35e-06 (1.349e-06, 1.352e-06)

Goodness of fit:
SSE: 1.26e+06
R-square: 1
Adjusted R-square: 1
RMSE: 424.3

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



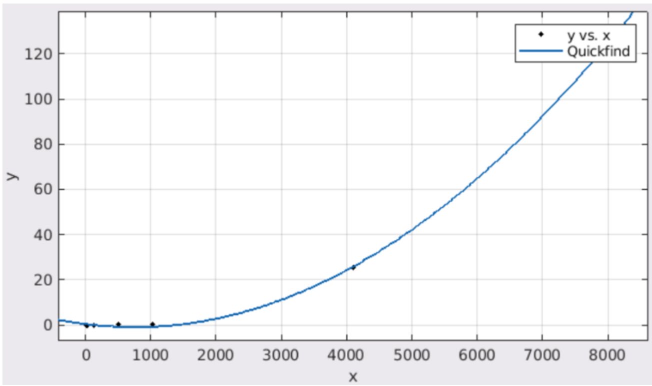
General model:
f(x) = a*x^2*log2(x)+b
Coefficients (with 95% confidence bounds):
a = 6.838e-06 (6.764e-06, 6.912e-06)
b = 13.06 (-10.97, 37.08)

Goodness of fit:
SSE: 3477
R-square: 0.9999
Adjusted R-square: 0.9999
RMSE: 24.07

Suppose that f(N) = $6.838*10^{-6}*N^2*\log(N)+13.06$.

Assume that g(N) = $N^2*\log(N)$ and c =7. To get f(N) < c*g(N) (N>Nc), we can set Nc = 10.

3. Quick find



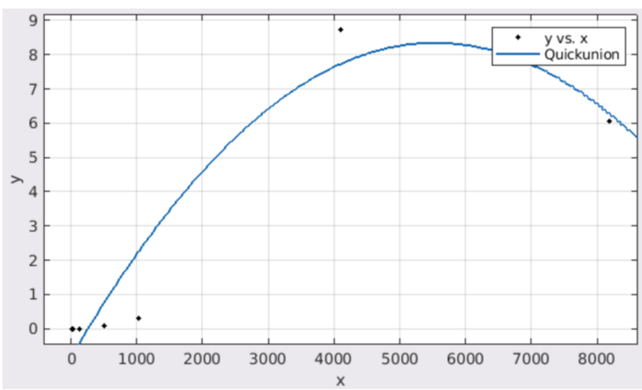
Linear model Poly2:
f(x) = p1*x^2 + p2*x + p3
Coefficients (with 95% confidence bounds):
p1 = 2.401e-06 (2.236e-06, 2.566e-06)
p2 = -0.003679 (-0.005013, -0.002345)
p3 = 0.6929 (-0.5789, 1.965)

Goodness of fit:
SSE: 3.127
R-square: 0.9998
Adjusted R-square: 0.9997
RMSE: 0.8841

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



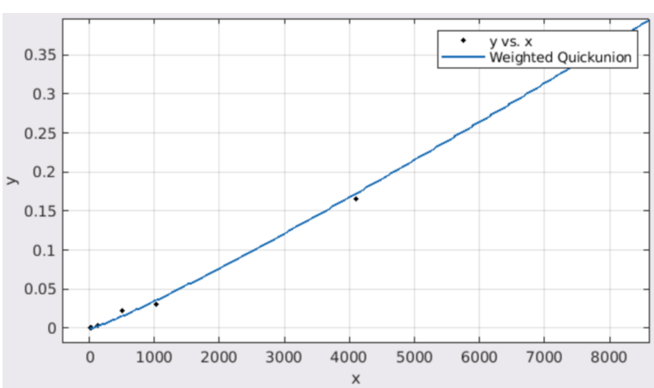
Linear model Poly2:
f(x) = p1*x^2 + p2*x + p3
Coefficients (with 95% confidence bounds):
p1 = -2.986e-07 (-5.388e-07, -5.842e-08)
p2 = 0.003311 (0.001367, 0.005255)
p3 = -0.8164 (-2.669, 1.037)

Goodness of fit:
SSE: 6.637
R-square: 0.917
Adjusted R-square: 0.8755
RMSE: 1.288

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



General model:
f(x) = a*x*log2(x)+b
Coefficients (with 95% confidence bounds):
a = 3.517e-06 (3.375e-06, 3.658e-06)
b = -0.0001949 (-0.00648, 0.006091)

Goodness of fit:
SSE: 0.0001459
R-square: 0.9988
Adjusted R-square: 0.9985
RMSE: 0.005401

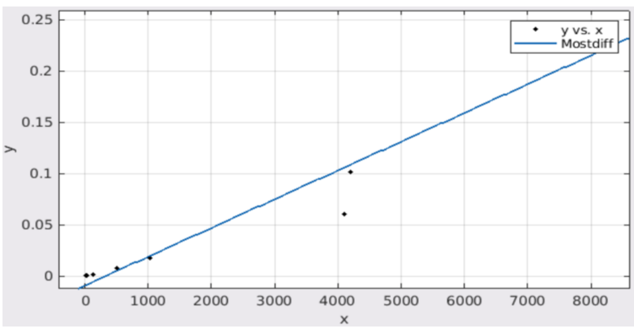
Suppose that f(N) = $3.517*10^{-6}$*N*log(N)-0.0001949.

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



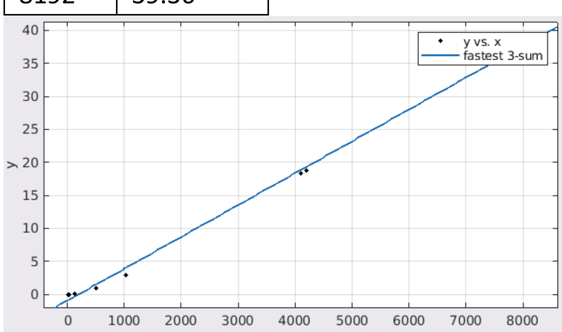
Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 2.805e-05 (2.108e-05, 3.501e-05)
p2 = -0.008749 (-0.03372, 0.01622)

Goodness of fit:
SSE: 0.002989
R-square: 0.9418
Adjusted R-square: 0.9321
RMSE: 0.02232

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



Linear model Poly1:
f(x) = p1*x + p2
Coefficients (with 95% confidence bounds):
p1 = 0.004831 (0.004571, 0.00509)
p2 = -0.9268 (-1.858, 0.004418)

Goodness of fit:
SSE: 4.156
R-square: 0.9971
Adjusted R-square: 0.9966
RMSE: 0.8323

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