

CSCE 313 Programming Assignment 1

Name: Yingjian Wang

UIN: 726004401

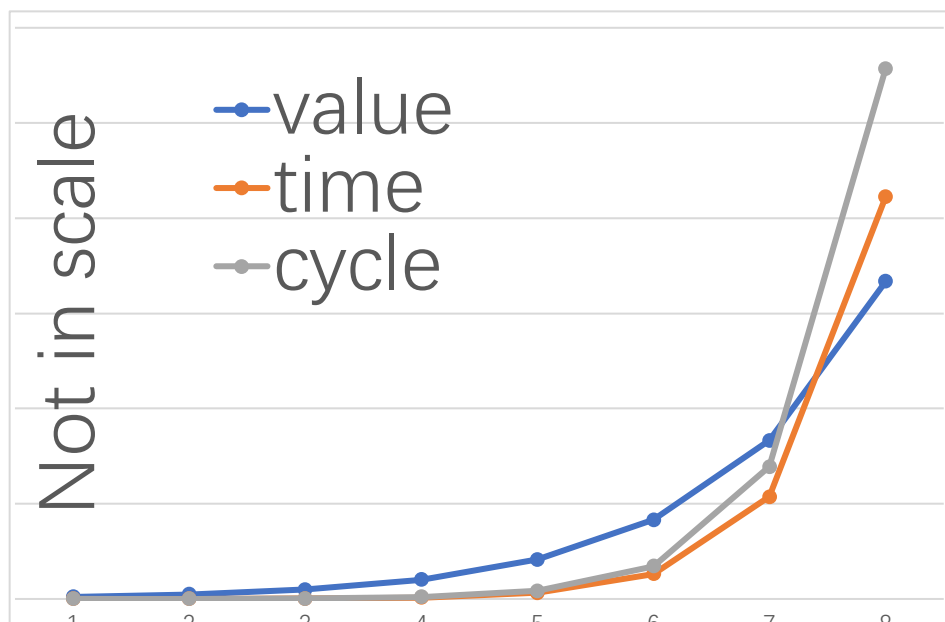
E-mail: wang19485@tamu.edu

Test Outcome :

(bbs = 128 bytes, tms = 128 * 1024 * 1024 bytes)

Ackermann Function: $A(n, m)$

m \ n=3	Time/s	Ackermann Value	allocate/free cycles
m=1	0.008623	13	106
m=2	0.037897	29	541
m=3	0.150609	61	2432
m=4	0.629331	125	10307
m=5	2.574050	253	42438
m=6	10.661755	509	172233
m=7	43.64596	1021	693964
m=8	171.874338	2045	2785999



Sample results: (bbs = 128, tms = 128 * 1024 * 1024)

```
=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 1
Ackerman(3, 1): 13
Time taken: [sec = 0, musec = 8623]
Number of allocate/free cycles: 106

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 2
Ackerman(3, 2): 29
Time taken: [sec = 0, musec = 37897]
Number of allocate/free cycles: 541

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 3
Ackerman(3, 3): 61
Time taken: [sec = 0, musec = 150609]
Number of allocate/free cycles: 2432
```

```
=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 4
Ackerman(3, 4): 125
Time taken: [sec = 0, musec = 629331]
Number of allocate/free cycles: 10307

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 5
Ackerman(3, 5): 253
Time taken: [sec = 2, musec = 573050]
Number of allocate/free cycles: 42438

=====
```

```

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 8
Ackerman(3, 8): 2045
Time taken: [sec = 171, musec = 876338]
Number of allocate/free cycles: 2785999

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 7
Ackerman(3, 7): 1021
Time taken: [sec = 43, musec = 64596]
Number of allocate/free cycles: 693964

=====
Please enter parameters n (<=3) and m (<=8) to ackerman function
Enter 0 for either n or m in order to exit.

n = 3
m = 6
Ackerman(3, 6): 509
Time taken: [sec = 10, musec = 661755]
Number of allocate/free cycles: 172233

```

As you can see, the increment of n and m increases the value of Ackermann exponentially. And the time and the number of cycles even increase more rapidly.

Compile the program:

1. Use command "make",
2. After successful compiling, write
"g++ BuddyAllocator.o Ackerman.o Main.o"
3. "/a.out -b 128 -s 134217728" to run the program, the numbers are available to change.

- **Important Notice:** the program can't accept value like "128 * 1024 * 1024" as command line argument, "134217728" is needed to run the original-data test
- The program cannot accept wrong format of command line argument!

Correct Examples:

```
/a.out -b 256 -s 67108864
```

```
/a.out -s 134217728
```

Wrong Examples:

```
/a.out [-b <256>] [-s <134217728>]
```

```
/a.out -b -s 134217728
```

- I would suggest that provide big total memory size on the command line argument if you want to let $n=3$, because the default total memory size (512 kb) is not enough when $n=3$. If you even provide much less total memory size, it's very likely that it can only pass the easy-test. But no worry, the program can catch these out of memory error and exits safely.

Analysis:

In my opinion, if the user requested an allocation for the data and didn't return. And then, the user keeps asking for more allocation of data, which will make the program spend more time to find available block of memory to allocate. If the user requested another allocation after they return the memory, the running time will reduce a lot.