HW8 Report

Name: Xiaowei Tan

Student ID: 69203272

Screenshots for Lab 8.2

Compile

```
$ make
echo     ------------compiling PairWithClosestProduct.cpp to create executable program main----------------
------------compiling PairWithClosestProduct.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   PairWithClosestProduct.cpp   -o   PairWithClosestProduct
xiaoweit@andromeda-27 10:04:47 ~/253p/hw/hw8/pair
$
```

Test 1

```
xiaoweit@andromeda-27 10:04:47 ~/253p/hw/hw8/pair
$ valgrind ./PairWithClosestProduct
==26575== Memcheck, a memory error detector
==26575== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==26575== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==26575== Command: ./PairWithClosestProduct
==26575==
count of numbers:5
the numbers:1 5 4 2 3
the target:13
the closest pair is 3 and 4
==26575==
==26575== HEAP SUMMARY:
==26575==     in use at exit: 72,704 bytes in 1 blocks
==26575==   total heap usage: 5 allocs, 4 frees, 72,764 bytes allocated
==26575==
==26575== LEAK SUMMARY:
==26575==    definitely lost: 0 bytes in 0 blocks
==26575==    indirectly lost: 0 bytes in 0 blocks
==26575==      possibly lost: 0 bytes in 0 blocks
==26575==    still reachable: 72,704 bytes in 1 blocks
==26575==         suppressed: 0 bytes in 0 blocks
==26575== Rerun with --leak-check=full to see details of leaked memory
==26575==
==26575== For counts of detected and suppressed errors, rerun with: -v
==26575== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:06:10 ~/253p/hw/hw8/pair
$
```

Test 2

```
$ valgrind ./PairWithClosestProduct
==26649== Memcheck, a memory error detector
==26649== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==26649== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==26649== Command: ./PairWithClosestProduct
==26649==
count of numbers:5
the numbers:-1 -5 -4 -2 -3
the target:11
the closest pair is -5 and -2
==26649==
==26649== HEAP SUMMARY:
==26649==     in use at exit: 72,704 bytes in 1 blocks
==26649==   total heap usage: 5 allocs, 4 frees, 72,764 bytes allocated
==26649==
==26649== LEAK SUMMARY:
==26649==    definitely lost: 0 bytes in 0 blocks
==26649==    indirectly lost: 0 bytes in 0 blocks
==26649==      possibly lost: 0 bytes in 0 blocks
==26649==    still reachable: 72,704 bytes in 1 blocks
==26649==         suppressed: 0 bytes in 0 blocks
==26649== Rerun with --leak-check=full to see details of leaked memory
==26649==
==26649== For counts of detected and suppressed errors, rerun with: -v
==26649== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:06:55 ~/253p/hw/hw8/pair
$
```

Test 3

```
$ valgrind ./PairWithClosestProduct
==26688== Memcheck, a memory error detector
==26688== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==26688== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==26688== Command: ./PairWithClosestProduct
==26688==
count of numbers:8
the numbers:-3 5 4 6 3 1 -4 -3
the target:7
the closest pair is 1 and 6
==26688==
==26688== HEAP SUMMARY:
==26688==     in use at exit: 72,704 bytes in 1 blocks
==26688==   total heap usage: 5 allocs, 4 frees, 72,764 bytes allocated
==26688==
==26688== LEAK SUMMARY:
==26688==    definitely lost: 0 bytes in 0 blocks
==26688==    indirectly lost: 0 bytes in 0 blocks
==26688==      possibly lost: 0 bytes in 0 blocks
==26688==    still reachable: 72,704 bytes in 1 blocks
==26688==         suppressed: 0 bytes in 0 blocks
==26688== Rerun with --leak-check=full to see details of leaked memory
==26688==
==26688== For counts of detected and suppressed errors, rerun with: -v
==26688== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:07:53 ~/253p/hw/hw8/pair
$
```

Test 4

```
$ valgrind ./PairWithClosestProduct
==26745== Memcheck, a memory error detector
==26745== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==26745== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==26745== Command: ./PairWithClosestProduct
==26745==
count of numbers:5
the numbers:-5 -2 0 1 3
the target:2
the closest pair is 1 and 3
==26745==
==26745== HEAP SUMMARY:
==26745==     in use at exit: 72,704 bytes in 1 blocks
==26745==   total heap usage: 5 allocs, 4 frees, 72,764 bytes allocated
==26745==
==26745== LEAK SUMMARY:
==26745==    definitely lost: 0 bytes in 0 blocks
==26745==    indirectly lost: 0 bytes in 0 blocks
==26745==      possibly lost: 0 bytes in 0 blocks
==26745==    still reachable: 72,704 bytes in 1 blocks
==26745==         suppressed: 0 bytes in 0 blocks
==26745== Rerun with --leak-check=full to see details of leaked memory
==26745==
==26745== For counts of detected and suppressed errors, rerun with: -v
==26745== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:08:55 ~/253p/hw/hw8/pair
$
```

Screenshots for Lab 8.3

Compile

```
$ make
echo     ------------compiling MaxAveSubtree.cpp to create executable program main----------------
------------compiling MaxAveSubtree.cpp to create executable program main----------------
g++  -ggdb   -std=c++11   MaxAveSubtree.cpp   -o   MaxAveSubtree
xiaoweit@andromeda-27 10:10:24 ~/253p/hw/hw8/maxave
$
```

Test1

```
xiaoweit@andromeda-27 10:39:47 ~/253p/hw/hw8/maxave
$ valgrind ./MaxAveSubtree
==30574== Memcheck, a memory error detector
==30574== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30574== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==30574== Command: ./MaxAveSubtree
==30574==
count of nodes:1
the nodes:1
the maximum average subtree is 0 and it has 1 nodes
==30574==
==30574== HEAP SUMMARY:
==30574==     in use at exit: 72,704 bytes in 1 blocks
==30574==   total heap usage: 3 allocs, 2 frees, 72,732 bytes allocated
==30574==
==30574== LEAK SUMMARY:
==30574==    definitely lost: 0 bytes in 0 blocks
==30574==    indirectly lost: 0 bytes in 0 blocks
==30574==      possibly lost: 0 bytes in 0 blocks
==30574==    still reachable: 72,704 bytes in 1 blocks
==30574==         suppressed: 0 bytes in 0 blocks
==30574== Rerun with --leak-check=full to see details of leaked memory
==30574==
==30574== For counts of detected and suppressed errors, rerun with: -v
==30574== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:39:58 ~/253p/hw/hw8/maxave
$
```

Test 2

```
$ valgrind ./MaxAveSubtree
==30630== Memcheck, a memory error detector
==30630== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30630== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==30630== Command: ./MaxAveSubtree
==30630==
count of nodes:5
the nodes:15 1 1 -1 -1
the maximum average subtree is 0 and it has 5 nodes
==30630==
==30630== HEAP SUMMARY:
==30630==     in use at exit: 72,704 bytes in 1 blocks
==30630==   total heap usage: 3 allocs, 2 frees, 72,844 bytes allocated
==30630==
==30630== LEAK SUMMARY:
==30630==    definitely lost: 0 bytes in 0 blocks
==30630==    indirectly lost: 0 bytes in 0 blocks
==30630==      possibly lost: 0 bytes in 0 blocks
==30630==    still reachable: 72,704 bytes in 1 blocks
==30630==         suppressed: 0 bytes in 0 blocks
==30630== Rerun with --leak-check=full to see details of leaked memory
==30630==
==30630== For counts of detected and suppressed errors, rerun with: -v
==30630== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:40:34 ~/253p/hw/hw8/maxave
$
```

Test 3

```
$ valgrind ./MaxAveSubtree
==30706== Memcheck, a memory error detector
==30706== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30706== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==30706== Command: ./MaxAveSubtree
==30706==
count of nodes:5
the nodes:1 2 3 20000 30000
the maximum average subtree is 4 and it has 1 nodes
==30706==
==30706== HEAP SUMMARY:
==30706==     in use at exit: 72,704 bytes in 1 blocks
==30706==   total heap usage: 3 allocs, 2 frees, 72,844 bytes allocated
==30706==
==30706== LEAK SUMMARY:
==30706==    definitely lost: 0 bytes in 0 blocks
==30706==    indirectly lost: 0 bytes in 0 blocks
==30706==      possibly lost: 0 bytes in 0 blocks
==30706==    still reachable: 72,704 bytes in 1 blocks
==30706==         suppressed: 0 bytes in 0 blocks
==30706== Rerun with --leak-check=full to see details of leaked memory
==30706==
==30706== For counts of detected and suppressed errors, rerun with: -v
==30706== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:41:10 ~/253p/hw/hw8/maxave
$
```

Test 4

```
$ valgrind ./MaxAveSubtree
==30805== Memcheck, a memory error detector
==30805== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30805== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==30805== Command: ./MaxAveSubtree
==30805==
count of nodes:5
the nodes:1 8 -3 -1 -2
the maximum average subtree is 1 and it has 3 nodes
==30805==
==30805== HEAP SUMMARY:
==30805==     in use at exit: 72,704 bytes in 1 blocks
==30805==   total heap usage: 3 allocs, 2 frees, 72,844 bytes allocated
==30805==
==30805== LEAK SUMMARY:
==30805==    definitely lost: 0 bytes in 0 blocks
==30805==    indirectly lost: 0 bytes in 0 blocks
==30805==      possibly lost: 0 bytes in 0 blocks
==30805==    still reachable: 72,704 bytes in 1 blocks
==30805==         suppressed: 0 bytes in 0 blocks
==30805== Rerun with --leak-check=full to see details of leaked memory
==30805==
==30805== For counts of detected and suppressed errors, rerun with: -v
==30805== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:41:38 ~/253p/hw/hw8/maxave
$
```

Test 5

```
$ valgrind ./MaxAveSubtree
==30867== Memcheck, a memory error detector
==30867== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==30867== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==30867== Command: ./MaxAveSubtree
==30867==
5count of nodes:
the nodes:5 5 5 5 5
the maximum average subtree is 4 and it has 1 nodes
==30867==
==30867== HEAP SUMMARY:
==30867==     in use at exit: 72,704 bytes in 1 blocks
==30867==   total heap usage: 3 allocs, 2 frees, 72,844 bytes allocated
==30867==
==30867== LEAK SUMMARY:
==30867==    definitely lost: 0 bytes in 0 blocks
==30867==    indirectly lost: 0 bytes in 0 blocks
==30867==      possibly lost: 0 bytes in 0 blocks
==30867==    still reachable: 72,704 bytes in 1 blocks
==30867==         suppressed: 0 bytes in 0 blocks
==30867== Rerun with --leak-check=full to see details of leaked memory
==30867==
==30867== For counts of detected and suppressed errors, rerun with: -v
==30867== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:42:16 ~/253p/hw/hw8/maxave
$
```

Screenshots for Lab 8.4

Compile

```
$ make
echo     ------------compiling LongestKDistnctSubstr.cpp to create executable program main--
--------------
------------compiling LongestKDistnctSubstr.cpp to create executable program main----------
-----
g++  -ggdb   -std=c++11   LongestKDistnctSubstr.cpp   -o   LongestKDistnctSubstr
xiaoweit@andromeda-27 10:55:10 ~/253p/hw/hw8/substr
$
```

Test 1

```
$ valgrind ./LongestKDistnctSubstr
==32111== Memcheck, a memory error detector
==32111== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==32111== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==32111== Command: ./LongestKDistnctSubstr
==32111==
string: abcbbbbcccbdddadacb
K: 2
the longest substring: bcbbbbcccb
==32111==
==32111== HEAP SUMMARY:
==32111==     in use at exit: 72,704 bytes in 1 blocks
==32111==   total heap usage: 9 allocs, 8 frees, 73,015 bytes allocated
==32111==
==32111== LEAK SUMMARY:
==32111==    definitely lost: 0 bytes in 0 blocks
==32111==    indirectly lost: 0 bytes in 0 blocks
==32111==      possibly lost: 0 bytes in 0 blocks
==32111==    still reachable: 72,704 bytes in 1 blocks
==32111==         suppressed: 0 bytes in 0 blocks
==32111== Rerun with --leak-check=full to see details of leaked memory
==32111==
==32111== For counts of detected and suppressed errors, rerun with: -v
==32111== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:57:00 ~/253p/hw/hw8/substr
$
```

Test 2

```
$ valgrind ./LongestKDistnctSubstr
==32013== Memcheck, a memory error detector
==32013== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==32013== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==32013== Command: ./LongestKDistnctSubstr
==32013==
string: 123456789
K: 5
the longest substring: 12345
==32013==
==32013== HEAP SUMMARY:
==32013==     in use at exit: 72,704 bytes in 1 blocks
==32013==   total heap usage: 10 allocs, 9 frees, 73,064 bytes allocated
==32013==
==32013== LEAK SUMMARY:
==32013==    definitely lost: 0 bytes in 0 blocks
==32013==    indirectly lost: 0 bytes in 0 blocks
==32013==      possibly lost: 0 bytes in 0 blocks
==32013==    still reachable: 72,704 bytes in 1 blocks
==32013==         suppressed: 0 bytes in 0 blocks
==32013== Rerun with --leak-check=full to see details of leaked memory
==32013==
==32013== For counts of detected and suppressed errors, rerun with: -v
==32013== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 10:56:04 ~/253p/hw/hw8/substr
$
```

Test 3

```
$ valgrind ./LongestKDistnctSubstr
==6950== Memcheck, a memory error detector
==6950== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==6950== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==6950== Command: ./LongestKDistnctSubstr
==6950==
string: 666666666
K: 5
the longest substring: 666666666
==6950==
==6950== HEAP SUMMARY:
==6950==     in use at exit: 72,704 bytes in 1 blocks
==6950==   total heap usage: 2 allocs, 1 frees, 72,744 bytes allocated
==6950==
==6950== LEAK SUMMARY:
==6950==    definitely lost: 0 bytes in 0 blocks
==6950==    indirectly lost: 0 bytes in 0 blocks
==6950==      possibly lost: 0 bytes in 0 blocks
==6950==    still reachable: 72,704 bytes in 1 blocks
==6950==         suppressed: 0 bytes in 0 blocks
==6950== Rerun with --leak-check=full to see details of leaked memory
==6950==
==6950== For counts of detected and suppressed errors, rerun with: -v
==6950== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 11:58:14 ~/253p/hw/hw8/substr
$
```

Test 4

```
$ valgrind ./LongestKDistnctSubstr
==7074== Memcheck, a memory error detector
==7074== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==7074== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==7074== Command: ./LongestKDistnctSubstr
==7074==
string: 123456556678
K: 5
the longest substring: 234565566
==7074==
==7074== HEAP SUMMARY:
==7074==     in use at exit: 72,704 bytes in 1 blocks
==7074==   total heap usage: 9 allocs, 8 frees, 73,024 bytes allocated
==7074==
==7074== LEAK SUMMARY:
==7074==    definitely lost: 0 bytes in 0 blocks
==7074==    indirectly lost: 0 bytes in 0 blocks
==7074==      possibly lost: 0 bytes in 0 blocks
==7074==    still reachable: 72,704 bytes in 1 blocks
==7074==         suppressed: 0 bytes in 0 blocks
==7074== Rerun with --leak-check=full to see details of leaked memory
==7074==
==7074== For counts of detected and suppressed errors, rerun with: -v
==7074== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 11:59:40 ~/253p/hw/hw8/substr
$
```

Test 5

```
$ valgrind ./LongestKDistnctSubstr
==7152== Memcheck, a memory error detector
==7152== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==7152== Using Valgrind-3.13.0 and LibVEX; rerun with -h for copyright info
==7152== Command: ./LongestKDistnctSubstr
==7152==
string: 12344321
K: 5
the longest substring: 12344321
==7152==
==7152== HEAP SUMMARY:
==7152==     in use at exit: 72,704 bytes in 1 blocks
==7152==   total heap usage: 5 allocs, 4 frees, 72,864 bytes allocated
==7152==
==7152== LEAK SUMMARY:
==7152==    definitely lost: 0 bytes in 0 blocks
==7152==    indirectly lost: 0 bytes in 0 blocks
==7152==      possibly lost: 0 bytes in 0 blocks
==7152==    still reachable: 72,704 bytes in 1 blocks
==7152==         suppressed: 0 bytes in 0 blocks
==7152== Rerun with --leak-check=full to see details of leaked memory
==7152==
==7152== For counts of detected and suppressed errors, rerun with: -v
==7152== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
xiaoweit@andromeda-27 12:00:24 ~/253p/hw/hw8/substr
$
```