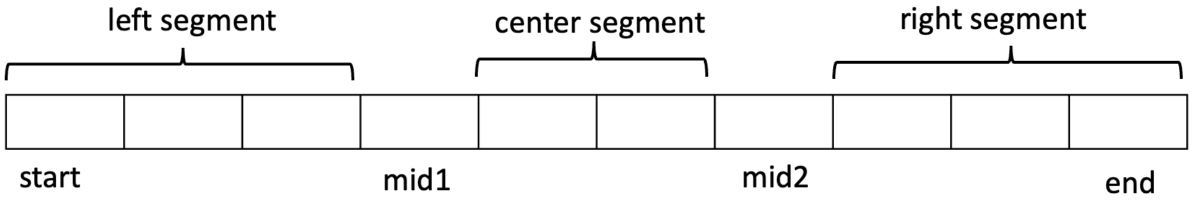
CSC 1302, Spring 2025
LAB EXAM 1 (4 March 2025)
SUBMISSION on tinman.cs.gsu.edu: `sudo handin1302 e1 TernarySearch.py`

Implement Ternary Search

Ternary search is a searching algorithm like binary search but divides the search space into three parts instead of two, as illustrated in the figure below:



Write a **recursive** Python function that implements the ternary search algorithm. The input to the function is a list of integers, `nums`, sorted in **descending** order, and `target,` the number being searched. Additional inputs are `left` and `right`, the indices of the left and right end of the segment being searched. Here are some hints for your implementation:

- If `target` is equal to one of the midpoint numbers, the search is successful, and the function should return the corresponding index (`mid1` or `mid2`).
- Otherwise, the function should narrow down the search space to 'left segment' or 'center segment' or 'right segment' based on the value of `target`.
- If `target` is not found within `nums`, the function should return `-1`.

Ternary Search program (TernarySearch.py)

```
def ternarySearch(target, nums, left, right):
    pass
```

Driver program (Driver.py)

```
from TernarySearch import *

def main():
  nums = [443,339,333,231,202,17,11,9,8,7,6,5,4,1]
  for target in [11,111]:
    print(ternarySearch(target,nums,0,len(nums)-1))   # should print 6 and -1

  nums = [10*i for i in range(100,0,-1)]
  for target in [460,466]:
    print(ternarySearch(target,nums,0,len(nums)-1))   # should print 54 and -1

main()
```

The above code template can be obtained at: https://tinman.cs.gsu.edu/~raj/1302/sp25/e1/