

Comp 160 focuses more on algorithmic concepts rather than implementation details. As such, all of the assignments we give are text based (as opposed to code based). For fun we thought we could release a coding challenge: these assignments are designed for students to better understand all minor details of a particular algorithm.

We emphasize that these coding assignments are *optional*. Consider them as a *bonus* that should only be attempted once you have done the usual homework. To keep this balance between optional and still want you to attempt to do them, we introduce the following rules:

- Coding assignments will have half the weight of normal homework. That is, each homework is worth roughly 30 points, whereas coding assignments will be roughly 15.
- Points that you get in Coding assignments will be additional: at the end of the semester we will add up all of the points that you got from both coding and homework assignments. We will divide this sum by the total number of points that you can get **from homework assignments only** and use this ratio to determine your overall homework grade.

Coding assignments can be done in either C++ or Python, whichever language you feel most comfortable on. You are free to ask TAs for help in the assignment, but please understand that not all TAs are expert on both languages. You may have to go to different times depending on the depth of your question (if unable, simply make a private post in Piazza).

## 1 Introduction

In this coding assignment you will implement `RANDSELECT`: given an array containing  $n$  numbers, and a positive integer  $k \leq n$ , `RANDSELECT` returns the  $k$ -th smallest element.

Together with this document you should have starter code: `main.py`, `randSelect.py` and the counterpart starting code for C++ (note that C++ version has needs `Makefile` and similar additional files). The `main` file is a very simple file: it creates an array of 5 numbers and calls `randSelect` function (that is in file `randSelect`). You can modify it, but it's mostly for you to test our your function. File `randSelect` is a placeholder without code (it is up to you to implement it).

In section ??, you will find an overview of the assignment and a suggested road map. In section ??, you will find some tips for implementing `RANDSELECT` with code examples for both Python and C++.

## 2 Implementation details

`RANDSELECT` is a randomized algorithm. In each iteration, the algorithm selects an element at random to be the pivot, partitions the other elements into two groups (smaller and larger than the pivot) and then recurses in the corresponding group. For a challenge, you can instead implement the deterministic version of Selection.

### 2.1 Headers

If you want to do the assignment in Python, the function header should be as follows:

## 2.2 Printing your progress

```
1 def randSelect(array : List[int], rank : int) -> int:  
2   # TODO Implement randSelect
```

If you instead decide to use C++, the header should be:

```
1 int randSelect(std::vector<int> array, int rank) {  
2   // TODO Implement randSelect  
3 }
```

Note how we use the STL vector to avoid potential memory leak issues.

## 2.2 Printing your progress

In each iteration of the algorithm you must print the following two lines into screen:

Looking for value with rank <number> in the array:<linebreak><array>

Naturally, <number> should be replaced for the rank that we are looking for, <linebreak> is just a line break, and <array> should be the full array (see below for exact details on how to print the array).

The second line you print will help us track the behavior of your algorithm:

Selected <element> as the pivot; its rank is <rank>; Thus, we recurse on <direction>.<linebreak>

In this case, <element> should be one of the input values, <rank> should be the rank of the element we selected as pivot and <direction> should be **left** or **right** depending on the side that the algorithm needs to recurse on. Also, note that there is a single space after each ";" or ",".

An example execution of the algorithm would be:

```
1 Looking for value with rank 3 in the array:  
2 [ 3 4 5 1 2 ]  
3 Selected 1 as the pivot; its rank is 0; Thus, we recurse on right.  
4 Looking for value with rank 2 in the array:  
5 [ 3 4 5 2 ]  
6 Selected 5 as the pivot; its rank is 3; Thus, we recurse on left.  
7 Looking for value with rank 2 in the array:  
8 [ 3 4 2 ]  
9 Selected 3 as the pivot; its rank is 1; Thus, we recurse on right.  
10 Looking for value with rank 0 in the array:  
11 [ 4 ]  
12 Selected 4 as the pivot; its rank is 0; Thus, we recurse on nothing. We are done.
```

Note that we will not test your print output against something (since the algorithm is randomized). We just want to see that you are recursing correctly.

## 2.3 Recommendations

In the following we map out a suggested guideline for you to implement the assignment. Although not compulsory, we strongly encourage you to follow the guidelines below, as they will make your life easier.

### 2.3.1 Partitioning

A key component of Selection algorithm is the PARTITION procedure. Ideally, you would want this function to satisfy the following requirements:

- Input is a list (an array/vector depending on your language choice) and the index of your pivot
- This function rearranges the elements of the list so that values smaller (or equal) than the pivot are to the left of the pivot, then the pivot, and finally the values that are larger than the pivot
- In class we described how to implement this algorithm without using additional memory. Try to do it in a similar way
- The procedure should return the new index of the pivot.
- For safety of mind, add a few special cases to consider (what if  $k$  is too large or negative? what if the vector is empty? etc). We will not test those cases, but it is always nice to have resilient code

### 2.3.2 randSelect

Once you have **Partition** implemented as above, the code of **randSelect** should be relatively simple. randSelect needs to satisfy the following requirements:

- Input is an array and an integer specifying the rank of the number you want to find. **Rank is 0-based** (to make it easier to implement). **Our tests run on 0-based rank indexing**, so do keep this in mind.
- You probably want to make a trampoline function that accepts a few more parameters (say, where in the list of elements your target is, and so on).
- Output is an integer which is the element at the rank requested. This integer would correspond to the integer positioned at the rank if the array were to be sorted.
- Although it does not need to be identical to the RANDSELECT described in class, the algorithm should reflect what is taught in class.
- Do not forget to solve base cases of recursion!

## 3 Coding help

### 3.1 Python

Python has lots of array manipulation tools. We highlight a few that we think will be useful in your implementation.

**Printing an array** We will use Python's default format for printing an array. As such, you can print it with a single command as shown below:

```
1 ray = [1,2,3,4,5]
2 print(ray) # Expected [1,2,3,4,5]
```

**Extracting a portion of an array** Splicing an array is also fairly simple, we use python splice notation to help us out here.

```
1 ray = [1,2,3,4,5]
2 splicedRay = ray[0:2] # ray[start:end:stepSize]
3 print(splicedRay) # Expected [1,2]
4 print(ray[0:4:2]) # Expected [1,3]
5 print(ray[::-1]) # Expected [5,4,3,2,1]
```

## 3.2 C++

**Array length** We can find the length of an array in Python is  $O(1)$  time using the built-in method for length.

```
1 ray = [1,2,3,4,5]
2 print(len(ray)) # Expected 5
```

**Generating a random number** Here is one way to generate a random pivot.

```
1 import random
2 ray = [1,2,3,4,5]
3 # Generates a number from 0-4
4 pivot = random.randint(0, len(ray) - 1)
```

## 3.2 C++

Now we show the analogous functions for C++:

**Printing a vector** C++ does not have a default print function for vectors. To level the playing field, we've provided a print function for vectors in C++.

```
1 void printVector(std::vector<int> v) {
2     std::cout << "[ ";
3     for (int i : v)
4         std::cout << i << " ";
5     std::cout << "]\n";
6 }
```

A simple copy/paste should work. We still encourage you to do a human verification (pay special attention to spaces!).

**Extracting a portion of a vector** If you have a vector  $v$  and you wanted a part of  $v$  from  $i$  to  $j$  (where  $i \leq j$ ), here is how you would do it:

```
1 part = std::vector<int>(v.begin() + i, v.begin() + j);
```

**Vector length** You can find the length of a vector in constant time using the method `length()` or `size()`

**Generating a random number** When using C++ we suggest using function `rand()` in the `<cstdlib>` library:

```
1 vector<int> v{1, 2, 3, 4, 5};
2
3 // generates number between 0 and 4
4 int pivot = rand() % (int)v.size();
```

## 4 Grading

Since this is mostly an exercise meant to help settle ideas, grading will be relatively simple and will be based on the following items:

**Correctness** Is your code correct? We will run a few scripts on your code. For this part it is essential that you follow all reporting steps as described above (as you know, a missing comma can confuse the script and think that you did a major mistake).

## 4.1 Valid libraries

**Efficiency** Is your algorithm fast? Does it show an asymptotic linear runtime on the size of  $n$ ?

**Clear** Is your code easy to follow? Can we understand what your code is doing at a glance? For guidance, we recommend to Comp 15's coding style guide (available on that course's webpage), but in general our standards will be lower than that course.

If you satisfy the above requirements you will get a high grade.

**Note:** because we see this as a *bonus* assignment, grading on this exercise will not be as good as in the regular homework. This means that you will just get a numerical grade with little or no feedback. We believe that doing the assignment itself and seeing how the algorithm executes is an interesting learning exercise for you to do. Make sure to do some tests on your own to verify that your program works properly. We give you a slight incentive in terms of grade, but the main motivation for doing the assignment should be for knowledge. Also, this is the kind of coding question that you may end up seeing in job interviews.

## 4.1 Valid libraries

### 4.1.1 Python

You are free to use any Python Standard Library Function other than functions that have sorting capabilities. See the official Python3, documentations found [here](#). Please reach out to course staff if you think you want to use a function from a library, and you are not sure if you are allowed to.

### 4.1.2 C++

You are free to use any function other than any sorting function in `<iostream>`, `<cstdlib>`, and `<vector>`.

## 4.2 How to submit

Submissions are done via Gradescope. Note that you only have to provide either `randSelect.cpp` or `randSelect.py` file (depending on the language you chose for the implementation).