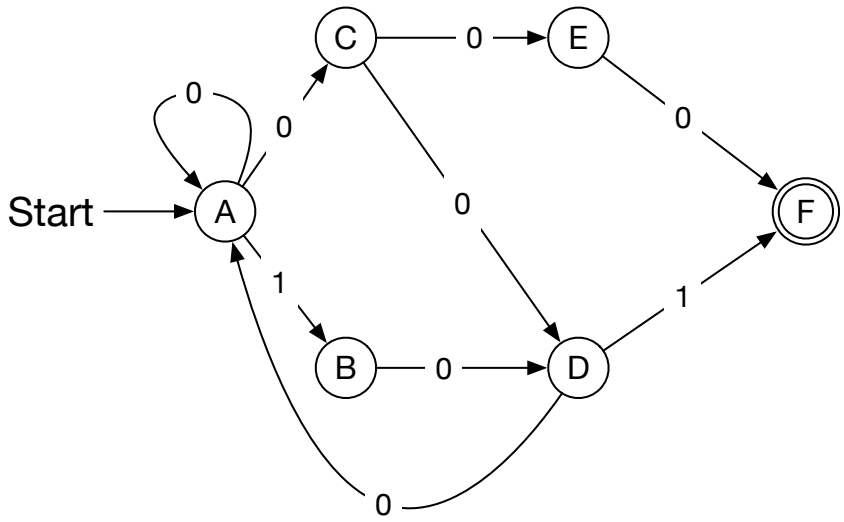
# Midterm Preparation

### NFA to DFA

Given the following NFA, determine the smallest string starting with a 0 and containing a 1 that can be accepted.




### Divide and Conquer Algorithm

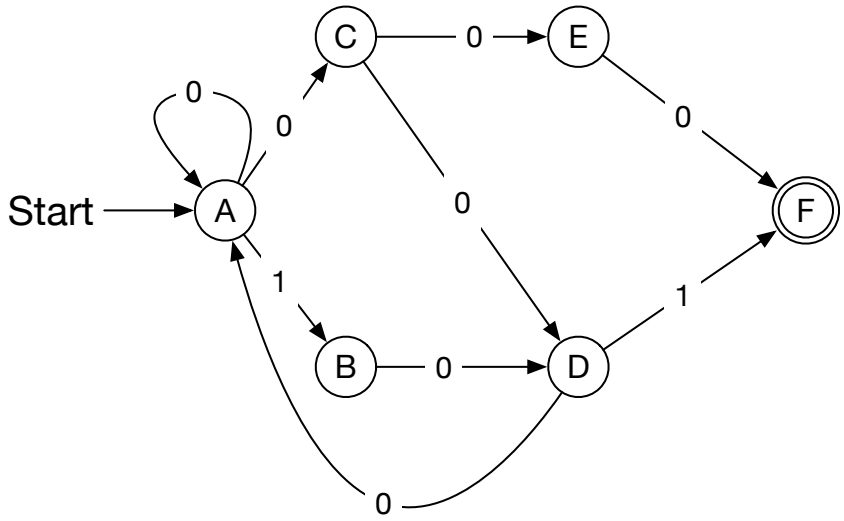
We are given a (very large) array of numbers containing the prices of a given stock. To test an AI algorithm for trading, we want to find the best interval for investing, were we buy the stock cheap and sell for a high prize. Let's call this the best run. Mathematically, we want to find $\max(\{a_j - a_i \mid i < j\})$.

Give a divide and conquer algorithm to calculate the longest run.  Hint: What are the possibilities for the longest run as regards to dividing the array into two halves?

# Solutions

### NFA to DFA

Any string accepted has to move the current state from A to F and this involves at least three steps. This means that each accepted string has at least three letters. The string "001" is accepted and therefore minimal.




### Divide and Conquer Algorithm

We divide the array into two halves, left and right. We then argue as follows: The best run could be (1) in left, could be (2) in right, or (3) it spans both, in which case it is between the minimum

```
def longest_run(array):
    if len(array) == 1:
        return 0, array[0], array[0]
    else:
        left = array[:len(array)//2]
        right = array[len(array)//2:]
        lbest, lmin, lmax = longest_run(left)
        rbest, rmin, rmax = longest_run(right)

        best = max(lbest, rbest, rmax-lmin)
        mymin = min(lmin, rmin)
        mymax = max(lmax, rmax)


        return best, mymin, mymax
```

on the left and the maximum on the right. For the recursive call, we return the value of the best run, the minimum, and the maximum. A Python implementation is given above.