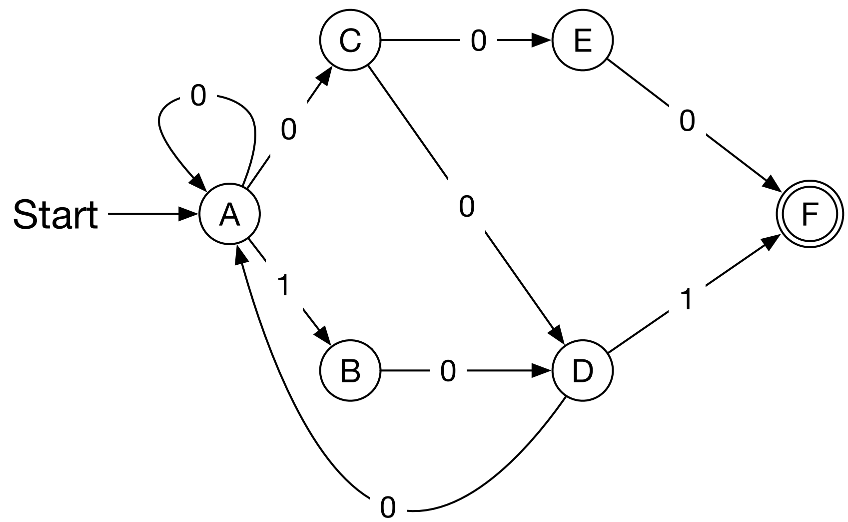
Midterm — Algorithms

1. NFA to DFA

Given the following NFA, find the transition of the state {A,C,E} on 0 and 1 in the corresponding DFA.



2. Asymptotic Notation:

Compare $\log(n)n$ and $n^{3/2}$. Give the result in terms of o, Θ, Ω.

3. 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 $\max(\{a_j - a_i \mid i < j\})$.

(a) What is the number of differences $\{a_j - a_i \mid i < j\}$? Let n be the length of the array.

(b) Alternatively, we look at a 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

```
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, 0)
        mymin = min(lmin, rmin)
        mymax = max(lmax, rmax)


        return best, mymin, mymax
```

(2) in right, or (3) it spans both, in which case it is between the minimum 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. **What is the recurrence relation for the runtime of this algorithm. Using the Master Theorem, what is the asymptotic runtime.**

4. Order Statistics

We have seen that by pairing the elements of an array we can simultaneously determine minimum and maximum of an array with $n = 2m + 1$ elements in $3m$ comparisons and of an array with $n = 2m$ elements with 3m-2 comparisons.

(a) Can we do better by grouping the array into larger groups? Assume the array has $n = 3l$ elements. We divide the array into l triplets with three elements each. We use three comparisons to determine the maximum and the minimum of each triplet. We then use $l - 1$ comparisons to find the maximum of the triplet maxima and the same number of comparisons to find the minimum of the triplet minima. How many comparisons will I use? Express the number in terms of n. Compare with using pairs.

(b) Now observe that in 1/3 of all cases, determining the maximum and the minimum of a triplet uses only two comparisons and in 2/3 of all cases, determining the maximum and the minimum of a triplet uses three comparisons. What is the expected number of comparisons in this case?

5. Knapsack

We are solving the 0-1 knapsack problem with a capacity of 24. There are ten objects, whose value and weight are given below.

	A	B	C	D	E	F	G	H	I	J
value	13	11	10	9	7	6	4	4	2	1
weight	7	6	5	5	4	3	3	2	1	1

The knapsack table is given below. Find the optimal selection of items.

	ϕ	A	A-B	A-C	A-D	A-E	A-F	A-G	A-H	A-I	A-J
0:	0	0	0	0	0	0	0	0	0	0	0
1:	0	0	0	0	0	0	0	0	0	2	2
2:	0	0	0	0	0	0	0	0	4	4	4
3:	0	0	0	0	0	0	6	6	6	6	6
4:	0	0	0	0	0	7	7	7	7	8	8
5:	0	0	0	10	10	10	10	10	10	10	10
6:	0	0	11	11	11	11	11	11	11	12	12
7:	0	13	13	13	13	13	13	13	14	14	14
8:	0	13	13	13	13	13	16	16	16	16	16
9:	0	13	13	13	13	17	17	17	17	18	18
10:	0	13	13	13	19	19	19	19	20	20	20
11:	0	13	13	21	21	21	21	21	21	22	22
12:	0	13	13	23	23	23	23	23	23	23	23
13:	0	13	24	24	24	24	25	25	25	25	25
14:	0	13	24	24	24	26	27	27	27	27	27
15:	0	13	24	24	24	28	29	29	29	29	29
16:	0	13	24	24	30	30	30	30	31	31	31
17:	0	13	24	24	32	32	32	32	33	33	33
18:	0	13	24	34	34	34	34	34	34	35	35
19:	0	13	24	34	34	34	36	36	36	36	36
20:	0	13	24	34	34	37	38	38	38	38	38
21:	0	13	24	34	34	39	40	40	40	40	40
22:	0	13	24	34	34	41	41	41	42	42	42
23:	0	13	24	34	43	43	43	43	44	44	44
24:	0	13	24	34	43	43	45	45	45	46	46