HW3

Ronan Wallace

February 28, 2019

Alphametic Program

Included in .zip folder

Section 3.1 p.103

8) Sort list E, X, A, M, P, L, E using selection sort

First, find the smallest letter (in this case, A) and switch to the front

E X **A** M P L E → **A** X E M P L E

A is now sorted. Exclude already sorted letters when finding the next smallest letter

X **E** M P L E → **E** X M P L E

A and E are now sorted. Exclude sorted letters when finding next smallest letter

X M P L **E** → **E** M P L X

A, E, and E are now sorted. Exclude when finding next letter

M P **L** X → **L** P M X

A, E, E, and L are now sorted. Exclude when finding next letter

P **M** X → **M** P X

A, E, E, L, and M are now sorted. Exclude when finding next letter

P X → **P** X

A, E, E, L, M, and P are now sorted. Exclude when finding next letter

X → **X**

A, E, E, L, M, P, and X are all sorted. The resulting list is:

A E E L M P X

Sorting Complete

11) Sort list E, X, A, M, P, L, E using bubble sort

E X A M P L E → **E X** A M P L E
E **X A** M P L E → E **A X** M P L E
E A **X M** P L E → E A **M X** P L E
E A M **X P** L E → E A M **P X** L E
E A M P **X L** E → E A M P **L X** E
E A M P L **X E** → E A M P L **E X**

X is in the correct position and does not need to be checked again.

E A M P L E X → **A E** M P L E X
A **E M** P L E X → A **E M** P L E X
A E **M P** L E X → A E **M P** L E X
A E M **P L** E X → A E M **L P** E X
A E M L **P E** X → A E M L **E P** X

P is in the correct position and does not need to be checked again.

A E M L E P X → **A E** M L E P X
A **E M** L E P X → A **E M** L E P X
A E **M L** E P X → A E **L M** E P X
A E L **M E** P X → A E L **E M** P X

M is in the correct position and does not need to be checked again.

A E L E M P X → **A E** L E M P X
A **E L** E M P X → A **E L** E M P X
A E **L E** M P X → A E **E L** M P X

L is in the correct position and does not need to be checked again.

A E E L M P X → **A E** E L M P X
A **E E** L M P X → A **E E** L M P X

All A, E, and E are in the correct position.

Sorting Complete

Section 3.4 p.120-121

1a) $\Theta(n!)$

Exhaustive-search: $n * \frac{1}{2}(n-1)!$

$\frac{1}{2}n! \leq 10,000,000,000 * t$

$n! \leq 2 * 10,000,000,000 * t$

1b.i) 1 hour = 3,600 seconds

$n! \leq 2 * 10,000,000,000 * 3,600$

$n! \leq 72,000,000,000,000$

$16! \leq 72,000,000,000,000$

The most that n can be in this equality is 16, therefore, the most cities that the problem can be solved in 1 hour is 16 cities.

1b.ii) 3,600 seconds * 24 = 86,400 seconds in 24 hours

$n! \leq 2 * 10,000,000,000 * 86,400$

$n! \leq 1,728,000,000,000,000$

$17! \leq 1,728,000,000,000,000$

The most that n can be in this equality is 17, therefore, the most cities that the problem can be solved in 24 hours is 17 cities.

1b.iii) 86,400 seconds * 365 = 31,536,000 seconds in a year

$n! \leq 2 * 10,000,000,000 * 31,536,000$

$n! \leq 630,720,000,000,000,000$

$19! \leq 630,720,000,000,000,000$

The most that n can be in this equality is 19, therefore, the most cities that the problem can be solved in 1 year is 19 cities.

1b.iv) 31,536,000 seconds * 100 = 31,536,000,000 seconds in a century

$n! \leq 2 * 10,000,000,000 * 31,536,000,000$

$n! \leq 630,720,000,000,000,000,000$

$21! \leq 630,720,000,000,000,000,000$

The most that n can be in this equality is 21, therefore, the most cities that the problem can be solved in 1 century is 21 cities.

9a)

$$\frac{64!}{56!} = 178,462,987,637,760 \text{ +different positions}$$

Estimated Time of Completion:

$$\frac{178,462,987,637,760}{10,000,000,000} \approx 17,846.299 \text{ seconds} \approx 4.957 \text{ hours}$$

9b)

$$64 * 56 * 48 * 40 * 32 * 24 * 16 * 8 = 676,457,349,120 \text{ different positions}$$

Estimated Time of Completion:

$$\frac{676,457,349,120}{10,000,000,000} \approx 67.646 \text{ seconds} \approx 1.127 \text{ minutes}$$

9c)

$$64 * 49 * 36 * 25 * 16 * 9 * 4 * 1 = 1,625,702,400 \text{ different positions}$$

Estimated Time of Completion:

$$\frac{1,625,702,400}{10,000,000,000} \approx 0.163 \text{ seconds}$$