12 Jun'25

→ Loops to find smallest number in a list

HW: Use this function to sort a list using the algorithm from the last class (using list operations - append, remove)

→ Without list operations: (append/remove) (like with arrays of fixed size)

Algorithm:

List - [2, 7, 9, 6, 4, 18]

→ Find smallest and put it at the beginning of the list

[2, 7, 9, 6, 4, 18]

→ Find smallest from the second onwards
→ swap with the second item

[2, 4, 9, 6, 7, 8]

Now the first two are sorted.

→ Repeat this until the end of the list.

→ See 24.2. SortedList code

→ This algorithm for sorting is called 'selection' sort as you are 'selecting' the smallest item one at a time

→ HW: Try writing this code by yourself

Binary Numbers: It's numbers with true/false, or zeros and ones (like booleans) but there are more number of digits.

Booleans have only one digit.

Any number in decimal system can be converted to a binary number.

Eg: $23_{\times 10} = 10111_{\times 2}$

$23_{\times 10}$ → decimal (base 10)

$10111_{\times 2}$ → binary (base 2)

Decimal number

$10)\ 2342\ 1\ (\ 2342$

$\quad\ \ 2442\ 0$

$\quad\ \ \ \ \ \ \ \ \ \ \ \ 1$ → Units digit = remainder when dividing with 10

$10)\ 2342\ (\ 234$

$\quad\ \ 234\ 0$

$\quad\ \ \ \ \ \ \ \ \ \ 2$ → 10s digit is remainder when dividing with 100 (or 10 twice)

$$23421 = 2\times10^4 + 3\times10^3 + 4\times10^2 + 2\times10^1 + 1\times10^0$$

If we want a number in binary system, we do the same thing using '2' in place of 10 — and find the remainders.

Eg:

$$\begin{array}{r} 2\,)\,23\,(\,11 \\ 22 \\ \hline 1 \\ \hline \end{array} \qquad \begin{array}{r} 2\,)\,11\,(\,5 \\ 10 \\ \hline 1 \\ \hline \end{array} \qquad \begin{array}{r} 2\,)\,5\,(\,2 \\ 4 \\ \hline 1 \\ \hline \end{array} \qquad \begin{array}{r} 2\,)\,2\,(\,1 \\ 2 \\ \hline 0 \\ \hline \end{array} \qquad \begin{array}{r} 2\,)\,1\,(\,0 \\ 0 \\ \hline 1 \\ \hline \end{array}$$

2^0 place, 2^1 place, 2^2 place, 2^3 place, 2^4 place

$$23_{\times10} = 10111_{\times2}$$

→ 10111_{x2} to convert to decimal number

$1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$

$= 16 + 0 + 4 + 2 + 1 = 23$

Similarly, you can do this with any base - not just 2 & 10.

Computers understand binary or base 2 better because it can use a lot of switches to represent 0's and 1's.

Base 10 system uses 10 symbols

0, 1, 2, ... 9

Base 2 system uses 2 symbols

0, 1

Base 'n' system will use 'n' symbols.

Base '16' or hexadecimal uses 16 symbols

0, 1, 2, 3 ... 9, A, B, C, D, E, F

Eg: $10FA942_{\times 16}$

$= (1\times16^6 + 0\times16^5 + F\times16^4 + A\times16^3 + 9\times16^2 + 4\times16^1 + 2\times16^0)_{\times 10}$

$F=15$, $A=10$

Eg: $25_{\times 9} = (2\times9 + 5\times1)_{\times 10} = (18+5)_{\times 10} = 23_{\times 10}$

→ HW: Practice all loop examples & sorting without looking at our code.

→ HW: For sorting, instead of ascending do descending order.