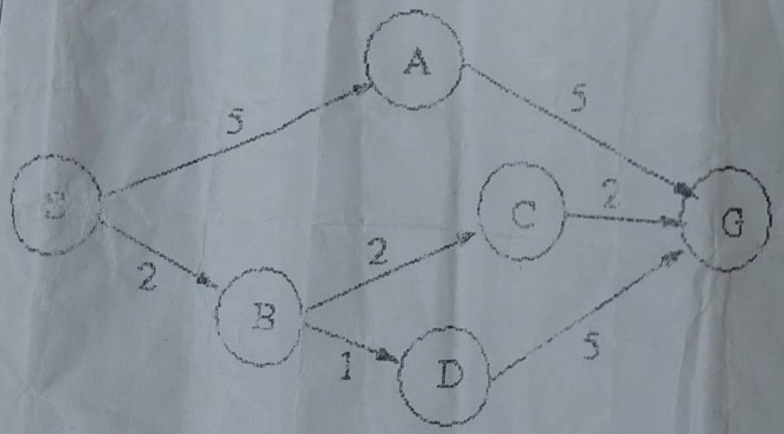
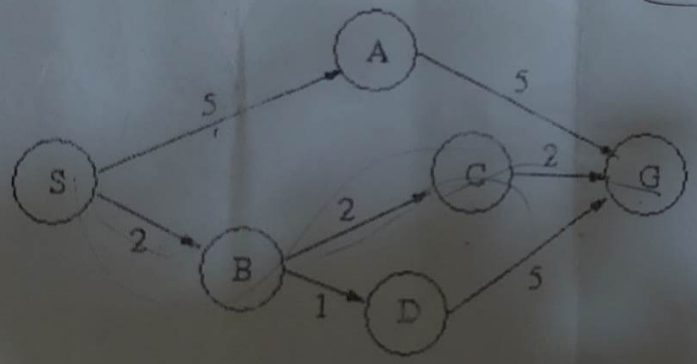
3.1. Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

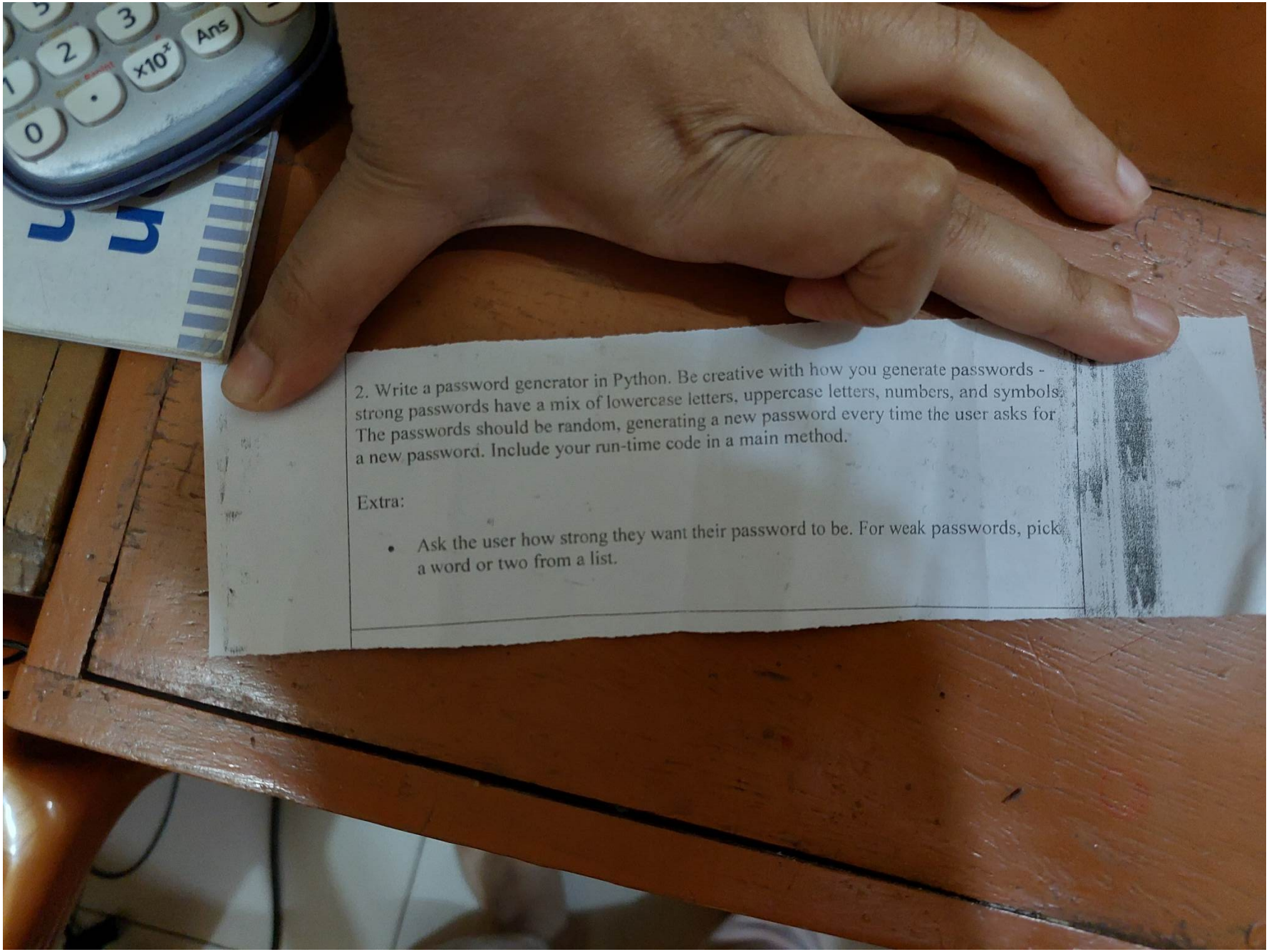
Give the three solution paths found by algorithm A using each of the three heuristic functions, respectively. Break ties alphabetically.

1. Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.

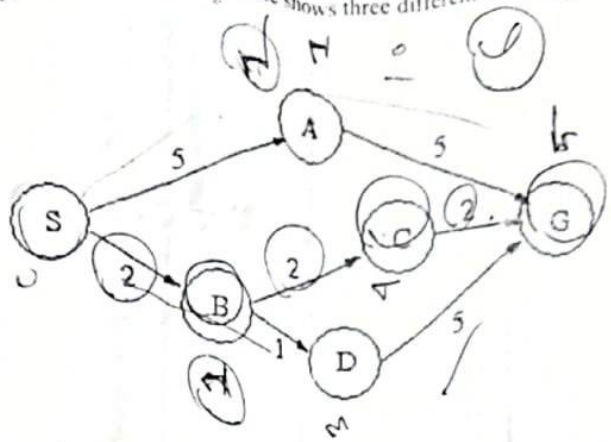


Node	*h1*	*h2*	*h3*
S	0	8	6
A	0	5	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

What solution path is found by Greedy Best-first search using *h2*? Break ties alphabetically.


2. Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password. Include your run-time code in a main method.
Extra:
• Ask the user how strong they want their password to be. For weak passwords, pick a word or two from a list.
Ans
×10ˣ

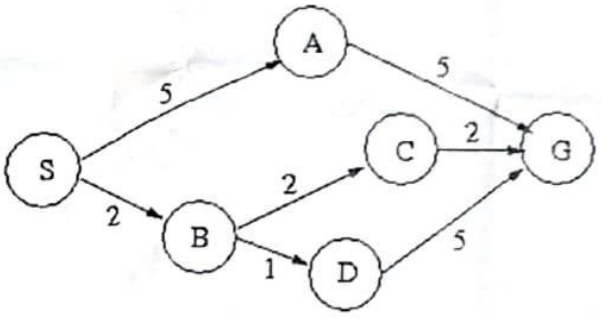
2. Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

What solution path is found by Uniform-Cost search? Break ties alphabetically.

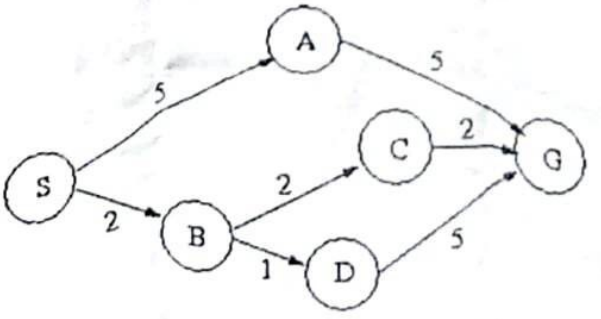
1. Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

What solution path is found by Greedy Best-first search using *h2*? Break ties alphabetically.

1. Consider the following search space where we want to find a path from the start state S to the goal state G. The table shows three different heuristic functions *h1*, *h2*, and *h3*.



Node	*h1*	*h2*	*h3*
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

What solution path is found by Greedy Best-first search using *h2*? Break ties alphabetically.

3. Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate boolean.

Extras:

- Use binary search.

3. Write a function that takes an ordered list of numbers (a list where the elements are in order from smallest to largest) and another number. The function decides whether or not the given number is inside the list and returns (then prints) an appropriate boolean.

Extras:

- Use binary search.

4. For this exercise, we will keep track of when our friend's birthdays are, and be able to find that information based on their name. Create a dictionary (in your file) of names and birthdays. When you run your program it should ask the user to enter a name, and return the birthday of that person back to them. The interaction should look something like this:

```
>>> Welcome to the birthday dictionary. We know the birthdays of:
Albert Einstein
Benjamin Franklin
Ada Lovelace
>>> Who's birthday do you want to look up?
Benjamin Franklin
>>> Benjamin Franklin's birthday is 01/17/1706.
```