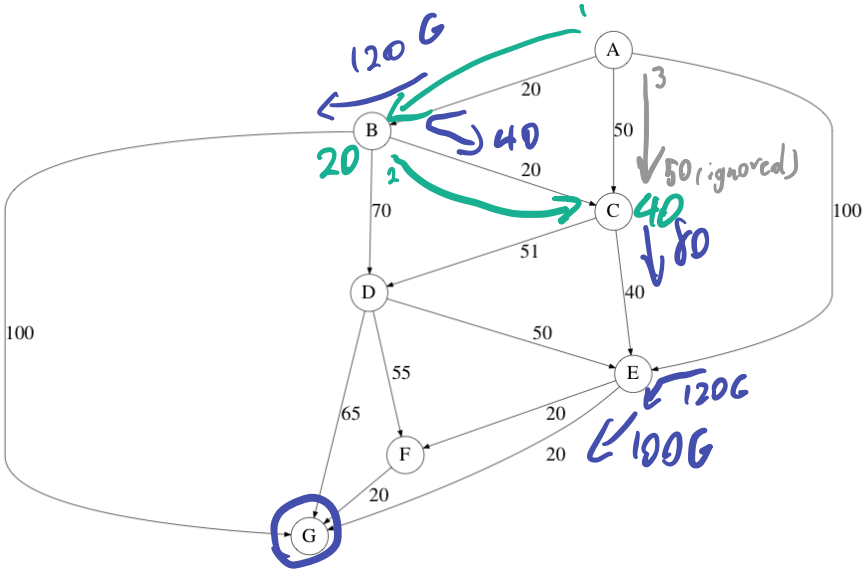

Node	h(n)
A	20
B	28
C	32
D	53
E	14
F	18
G	0

Problem 1 (Written; 10 pts): For the problem shown in Fig. 1, show that the heuristic is admissible ($h(n) \leq h^*(n)$ for all n). Note: You have to compute $h^*(n)$ for each n and compare to the $h(n)$ table.

Hint: It is best to work backwards from the goal, where $h^*(G) = 0$ (already at goal), $h^*(F) = 18$ (true minimum cost from (F) to (G)), $h^*(E) = 20$ (true minimum cost from (E) to (G)), etc.

$h(A) = 20 \leq 100 = h^*(A)$
$h(B) = 28 \leq 80 = h^*(B)$
$h(C) = 32 \leq 60 = h^*(C)$
$h(D) = 53 \leq 65 = h^*(D)$
$h(E) = 14 \leq 20 = h^*(E)$
$h(F) = 18 \leq 20 = h^*(F)$
$h(G) = 0 \leq 0 = h^*(G)$

A: 100 A→B→C→E→G
B: 80 B→C→E→G
C: 60 C→E→G
D: 65 D→G
E: 20 E→G
F: 20 F→G
G: 0 G

Problem 2 (Written; 20 pts): Manually conduct greedy best-first search on the graph below (Fig. 1), with initial node (A) and goal node (G). Actual cost from node to node are shown as edge labels. The heuristic function value for each node is shown in a spearate table to the right. Show:

1. Node list content at each step
2. Node visit order
3. Solution path
4. Cost of the final solution.

t = 0 Node List:
(A) h = 20, path = []
→ Visit (A) h = 20 (start point)

t = 0 Node List:
(E) h = 14, path = (A)
(B) h = 28, path = (A)
(C) h = 32, path = (A)
→ Visit (E) h = 14

t = 100 Node List:
(G) h = 0, path = (A)→(E)
(F) h = 18, path = (A)→(E)
(B) h = 28, path = (A)
(C) h = 32, path = (A)
→ Visit (G) h = 0
t = 120, Goal node found!

Node	h(n)
A	20
B	28
C	32
D	53
E	14
F	18
G	0

solution: A→E→G
cost: 120
expected: 20
cost = 6 × expected
$= \frac{6}{5} \times h^*(A)$

Problem 3 (Written; 20 pts): (1) Repeat the problem right above with A* search. (2) In addition, show the $f(n)$ value for all nodes expanded (you need this to sort them in the node list). (3)Which one gives a lower cost solution: Greedy best-first or A*?

t=0 Node List:

(A) h=20, path=[]

→ Visit (A) h=20

f=h+cost

t=0 Node List:

f=
14 (E) h=14, path=(A), cost=0
28 (B) h=28, path=(A), cost=0
32 (C) h=32, path=(A), cost=0

A→E → Visit (E) f=14

t=100 Node List: cost=100

(B) 28 h=28, path=(A), cost=0
(C) 32 h=32, path=(A), cost=0
(G) 100 h=0, path=(A)→(E), cost=100
(F) 118 h=18, path=(A)→(E), cost=100

A→B → Visit (B) f=28

t=120 Node List: cost=20

(G) 20 h=0, path=(A)→(B), cost=20
(C) 32 h=32, path=(A), cost=0
(C) 52 h=32, path=(A)→(B), cost=20
(D) 73 h=53, path=(A)→(B), cost=20
(G) 100 h=0, path=(A)→(E), cost=100
(F) 118 h=18, path=(A)→(E), cost=100

A→20→B → Visit (G) f=20

t=220 Node List: cost=100

(C) 32 h=32, path=(A), cost=0
(C) 52 h=32, path=(A)→(B), cost=20
(D) 73 h=53, path=(A)→(B), cost=20
(G) 100 h=0, path=(A)→(E), cost=100
(F) 118 h=18, path=(A)→(E), cost=100
exit (G) 120 h=0, path=(A)→(B)→(G), cost=120

A→C → Visit (C) f=32

t=270 Node List: cost=50

(C) 52 h=32, path=(A)→(B), cost=20
(F) 64 h=14, path=(A)→(C), cost=50
(D) 73 h=53, path=(A)→(B), cost=20
(G) 100 h=0, path=(A)→(E), cost=100
(D) 103 h=53, path=(A)→(C), cost=50
(F) 118 h=18, path=(A)→(E), cost=100
(G) 120 h=0, path=(A)→(B)→(G), cost=120

A→20→B → Visit (C) f=52

t=290 Node List: cost=20

(E) 54 h=14, path=(A)→(B)→(C), cost=40
(F) 64 h=14, path=(A)→(C), cost=50
(D) 73 h=53, path=(A)→(B), cost=20
(D) 93 h=53, path=(A)→(B)→(C), cost=40
(G) 100 h=0, path=(A)→(E), cost=100
(D) 103 h=53, path=(A)→(C), cost=50
(F) 118 h=18, path=(A)→(E), cost=100
(G) 120 h=0, path=(A)→(B)→(G), cost=120

A→20→B→20→C → Visit (E) f=54

cost=40

t=330

(F) 64 h=14, path=(A)→(C), cost=50
(D) 73 h=53, path=(A)→(B), cost=20
(G) 80 h=0, path=(A)→(B)→(C)→(E), cost=80 optimal path
(D) 93 h=53, path=(A)→(B)→(C), cost=40
(F) 98 h=18, path=(A)→(B)→(C)→(E), cost=80
(G) 100 h=0, path=(A)→(E), cost=100
(D) 103 h=53, path=(A)→(C), cost=50
(F) 118 h=18, path=(A)→(E), cost=100
(G) 120 h=0, path=(A)→(B)→(G), cost=120

A→50→C → Visit (E) f=64

t=370 cost=40

A→B→C→E 80 < A→C→E 90

do not expand

D 73 h=53, path = A→B, cost=20

G 80 h=0, path = A→B→C→E, cost=80

D 73 h=53, path = A→B→C, cost=40

F 98 h=18, path = A→B→C→E, cost=80

G 100 h=0, path = A→E, cost=100

D 103 h=53, path = A→C, cost=50

F 118 h=18, path = A→E, cost=100

G 120 h=0, path = A→B→G, cost=120

→ Visit D f=73

t=440 cost=70

G 80 h=0, path = A→B→C→E, cost=80

G 90 h=0, path = A→B→D, cost=90

D 73 h=53, path = A→B→C, cost=40

F 98 h=18, path = A→B→C→E, cost=80

G 100 h=0, path = A→E, cost=100

D 103 h=53, path = A→C, cost=50

F 108 h=18, path = A→B→D, cost=90

E 114 h=14, path = A→B→D, cost=90

F 118 h=18, path = A→E, cost=100

G 120 h=0, path = A→B→G, cost=120

→ Visit G f=80

t=460 cost=20

G 90 h=0, path = A→B→D, cost=90

D 73 h=53, path = A→B→C, cost=40

F 98 h=18, path = A→B→C→E, cost=80

priority to exit since we know ABCE < AE

G 100 h=0, path = A→B→C→E→G 100

G 100 h=0, path = A→E, cost=100

D 103 h=53, path = A→C, cost=50

F 108 h=18, path = A→B→D, cost=90

E 114 h=14, path = A→B→D, cost=90

F 118 h=18, path = A→E, cost=100

G 120 h=0, path = A→B→G, cost=120

→ Visit D f=80 DNE

cost=65 ABCEG 100 < 155 ABDG

t=525

D 73 h=53, path = A→B→C, cost=40

F 98 h=18, path = A→B→C→E, cost=80

G 100 h=0, path = A→B→C→E→G 100

G 100 h=0, path = A→E, cost=100

D 103 h=53, path = A→C, cost=50

F 108 h=18, path = A→B→D, cost=90

E 114 h=14, path = A→B→D, cost=90

F 118 h=18, path = A→E, cost=100

G 120 h=0, path = A→B→G, cost=120

G 155 h=0, path = A→B→D→G, cost=155

→ Visit D f=93 DNE

cost=51 ABD 90 < ABCD 91

t=576

→ Visit F f=98

cost=20

t=596

+ G 190 h=0, path = A→B→C→E→F, cost=100

→ Visit G f=100 DNE

cost=20 ABCEG 100 < ABCEFG 120

t=616

→ Visit G

t=636

Goal found

A→B→C→E→G

len: 100

t=636

A* found optimal solution

Greedy Best-fs did not

cost= is the same as g= in the code. cost= in the code is omniscient (knows h*) and I wanted A* to find the optimal path using f=h+(cost so far)