Q1/

Question (Map coloring with backtracking + arc consistency):

Use the backtracking algorithm with arc consistency to assign colors to five regions A, B, C, D, E so that no adjacent regions have the same color. Variables: A,B,C,D,E. Domains for all: {Red, Green, Blue}. Constraints: A≠B, A≠C, B≠E, C≠D, D≠E. Use alphabetical variable order (A→B→C→D→E) and try colors in the order Red, then Green, then Blue. First run arc consistency on initial domains before assigning A, and show the domains after each assignment.

Answer (showing domains at each step + final solution):

- **Initial arc-consistent domains (AC on all ≠ constraints):**

A={R,G,B}, B={R,G,B}, C={R,G,B}, D={R,G,B}, E={R,G,B} *(no pruning yet for ≠)*

- **Assign A = Red. Run AC:**

B cannot be Red ⇒ B={G,B}
C cannot be Red ⇒ C={G,B}
D={R,G,B}, E={R,G,B}

- **Assign B = Green (smallest in B′s domain). Run AC:**

E cannot be Green ⇒ E={R,B}
A fixed (R), C={G,B}, D={R,G,B}

- **Assign C = Green (smallest in C′s domain). Run AC:**

D cannot be Green ⇒ D={R,B}
A,B,E unchanged (A=R, B=G, E={R,B})

- **Assign D = Red (first in D′s domain). Run AC:**

E cannot be Red ⇒ E={B}

- **Assign E = Blue.** (singleton)
- **Final assignment (no backtracking needed):**

\boxed{A=Red,\; B=Green,\; C=Green,\; D=Red,\; E=Blue}
Checks: A≠B (R≠G), A≠C (R≠G), B≠E (G≠B), C≠D (G≠R), D≠E (R≠B) ✓

Q2/

Question (Search – DFS, BFS, UCS):

Consider the directed state-transition graph shown (S start, G goal; edge costs on edges). Ties are broken alphabetically and each state is expanded only once. (From the figure: outgoing edges are S→A(3), S→C(3), S→E(2), S→D(4); A→B(1); D→E(5); C→G(4); E→G(4).)

Depth-first search (DFS):

- **States expanded:** S,\;A,\;B,\;C,\;G
- **Path returned:** \boxed{S \rightarrow C \rightarrow G}

Breadth-first search (BFS):

- **States expanded:** S,\;A,\;C,\;D,\;E,\;G
- **Path returned (fewest steps):** \boxed{S \rightarrow C \rightarrow G}

Uniform-cost search (UCS):

- **States expanded (by increasing** g**-cost):** S,\;E,\;A,\;C,\;B,\;D,\;G
- **Path returned (least cost):** \boxed{S \rightarrow E \rightarrow G}
- **Path cost:** 2+4=\boxed{6}

(Why: UCS reaches G first via E with total cost 6; C→G would cost 3+4=7.)