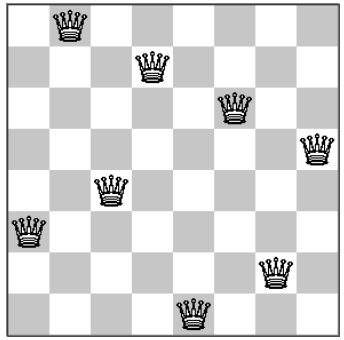
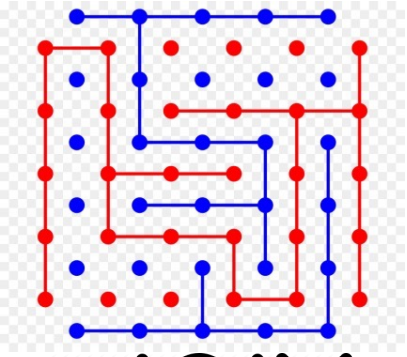
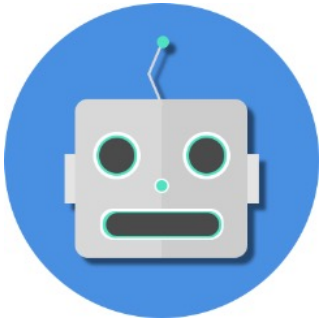
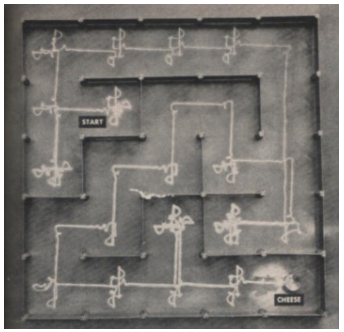
Artificial Intelligence:

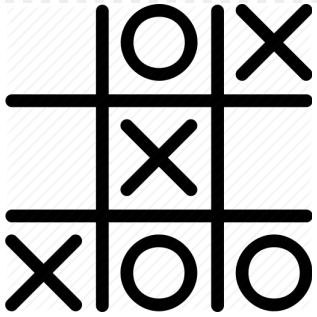
Past, Present and Future









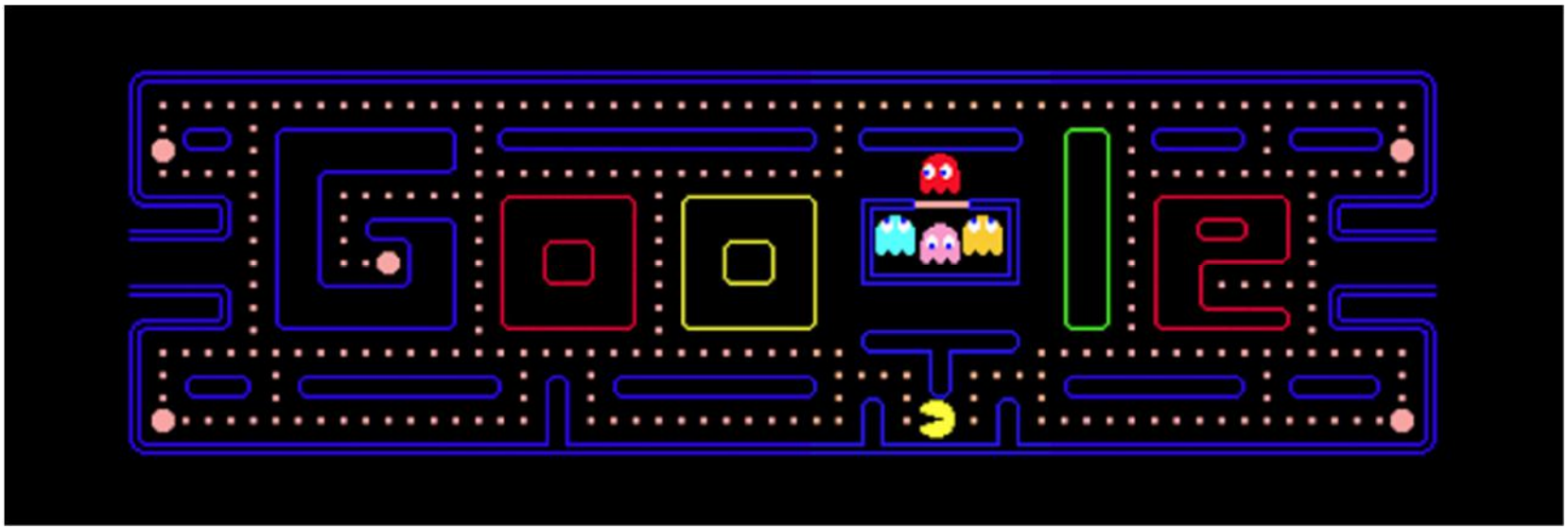


Chee Wei Tan

Maze Games

A popular classic game released by Namco in 1980

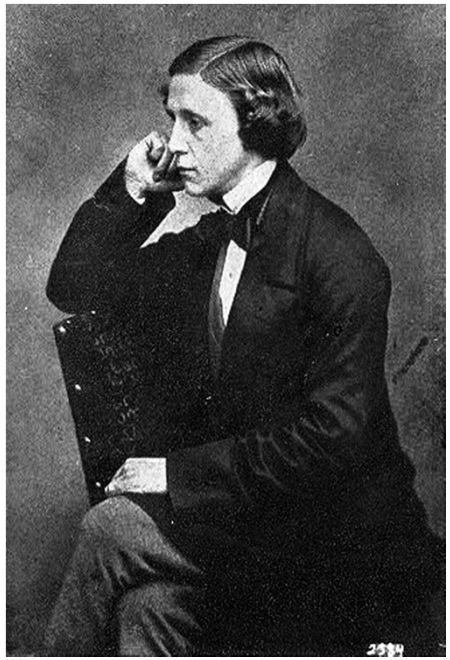
The player navigates Pac-Man through a maze with no dead ends. The objective of the game is to accumulate as many points as possible by eating dots, fruits, and blue ghosts. When all of the dots in a stage are eaten, that stage is completed, and the player will advance to the next.



Type Pac Man in Google search bar and play!

Word Ladder Game

A word ladder puzzle begins with two words, and to solve the puzzle one must find a chain of other words to link the two, in which two adjacent words (that is, words in successive steps) differ by one letter. Lewis Carroll invented the game on Christmas day in 1877



https://en.wikipedia.org/wiki/Lewis_Carroll

Word Ladder Game

From HEAD to TAIL:

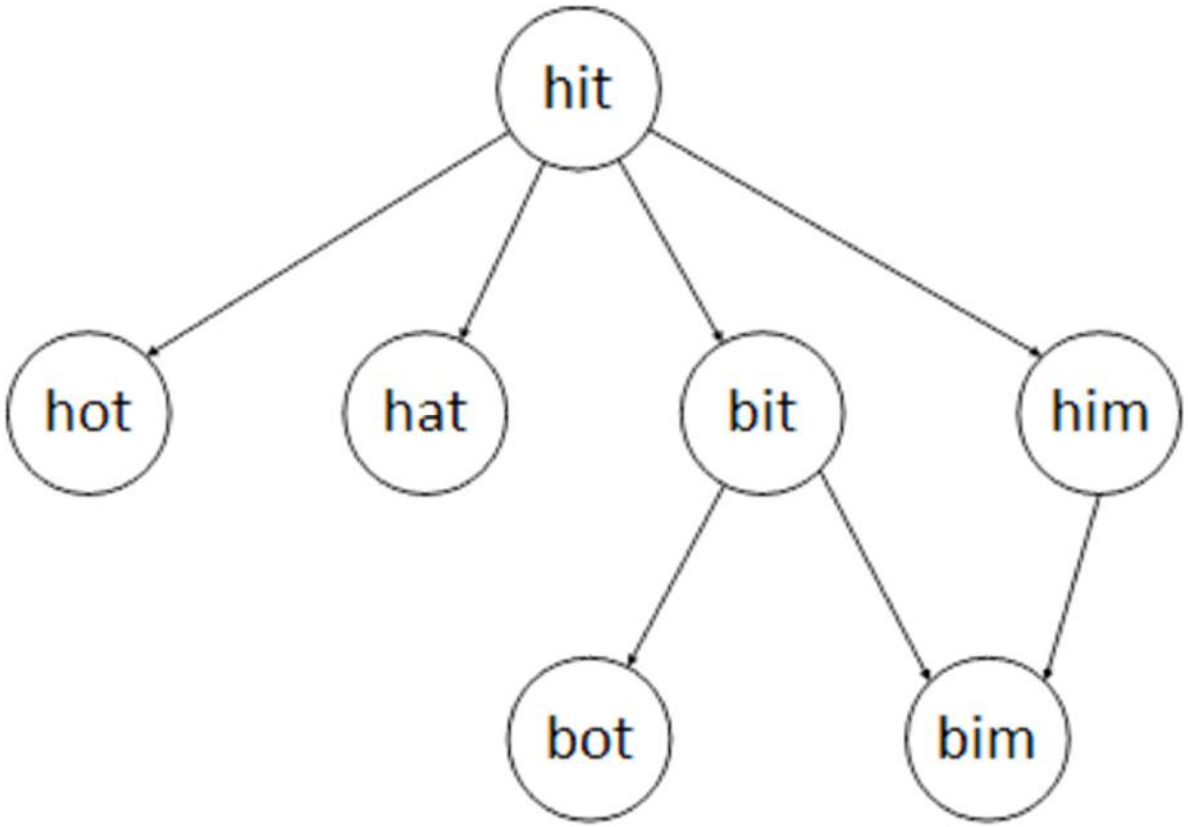
H E A D → H E A L → T E A L → T E L L → T A L L → T A I L

Five moves needed. Can you come up with fewer moves? How many possible solutions?

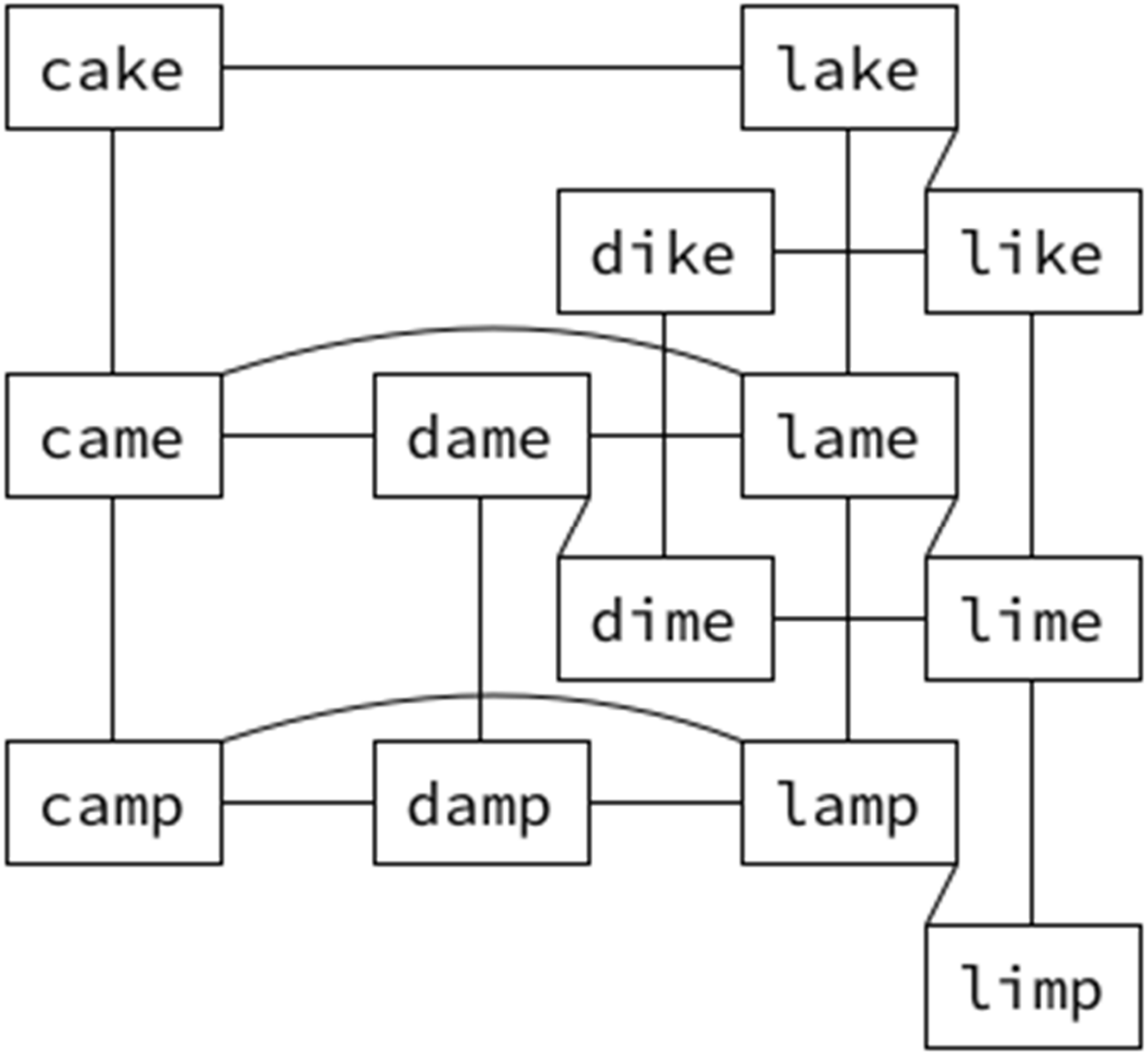
APE to MAN

Graph of Word Ladder Game

A graph is an important mathematical object:
Graph vertex or node: model a state of a game
Graph edge: model transition or relationship

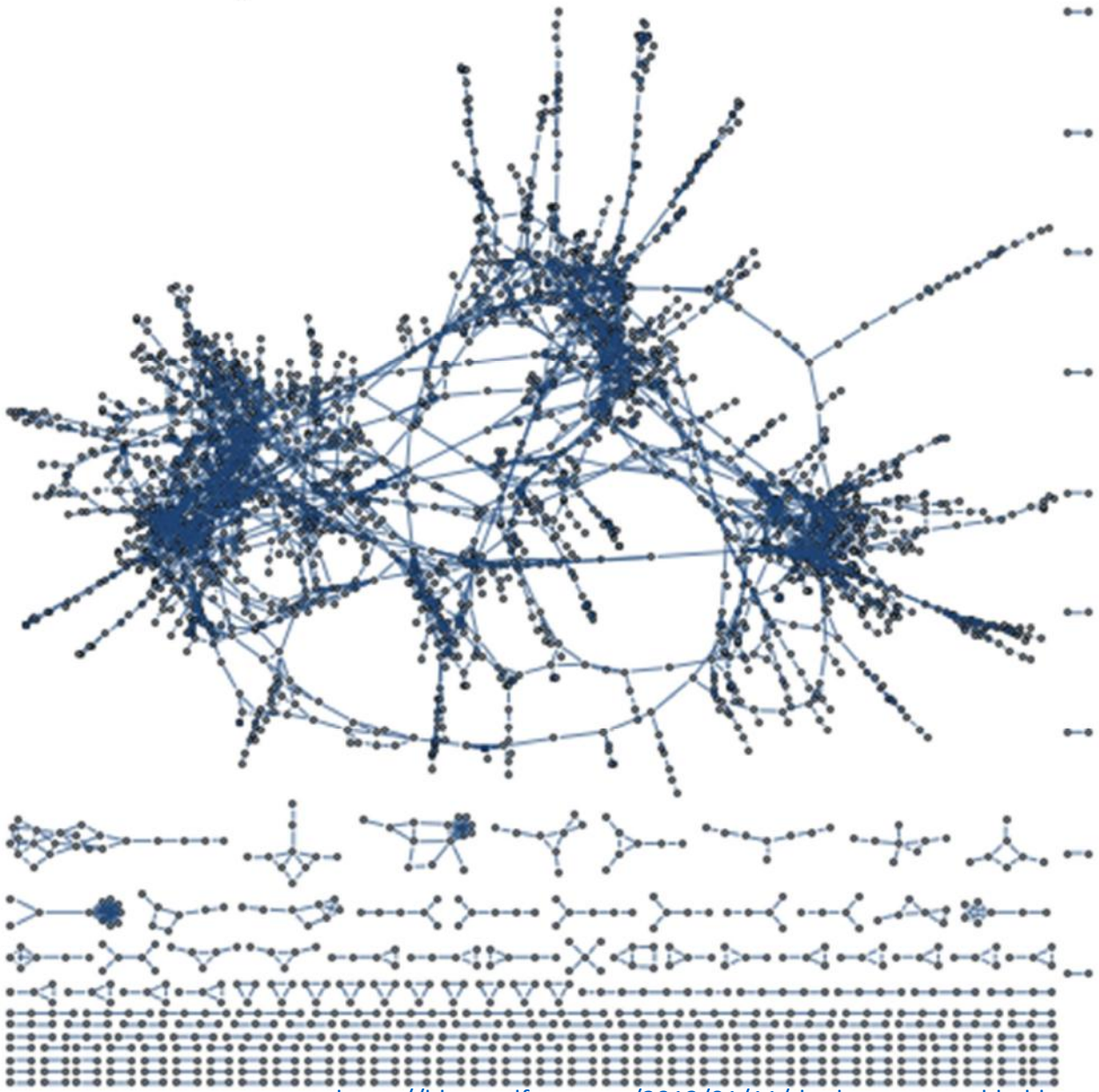


Graph of Word Ladder Game



Graph of Word Ladder Game

Out[4]=



[Donald Knuth](#) investigated Word Ladder game in 60's and found 817 five-letter words that are *aloof* (completely unconnected). Can you find any of these?

https://blog.wolfram.com/2012/01/11/the-longest-word-ladder-puzzle-ever

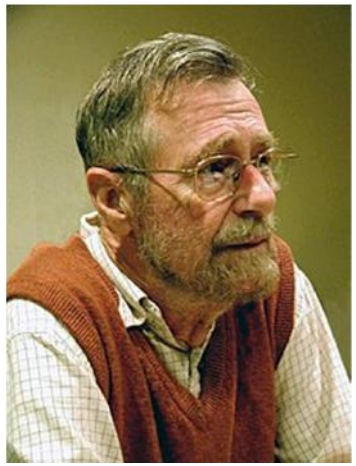
Backward Reasoning

- Imagine you have already solved the problem you are trying to solve. Work backwards from your solution to the starting point of your problem. Backward reasoning is also known as backward induction in mathematics or retrograde analysis in chess.
- Working backwards is suitable for problems, where some information has not been provided at the beginning of the problem. It helps to start with the answer and work methodically backwards to fill in the missing information. You can even uncover new solution through this process.
- This strategy is useful in dealing with problems that require a sequence of decisions to be made (multi-stage decision process). The decisions occur one after the other and each stage is affected by what comes next and what was decided before.

Maze-Solving: Graph Algorithms

- Maze-solving is a task of finding a desirable path from a given vertex to another desired vertex in a graph
- The *shortest path* may be desirable due to modeling considerations such as costs, efficiency or demonstrating skills
- Given a graph and two vertices (start and finishing), finding the *shortest path* from one to the other was conceived in 1956 by Dijkstra while giving a computer demo
- Backward reasoning underlies Dijkstra's algorithm

https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

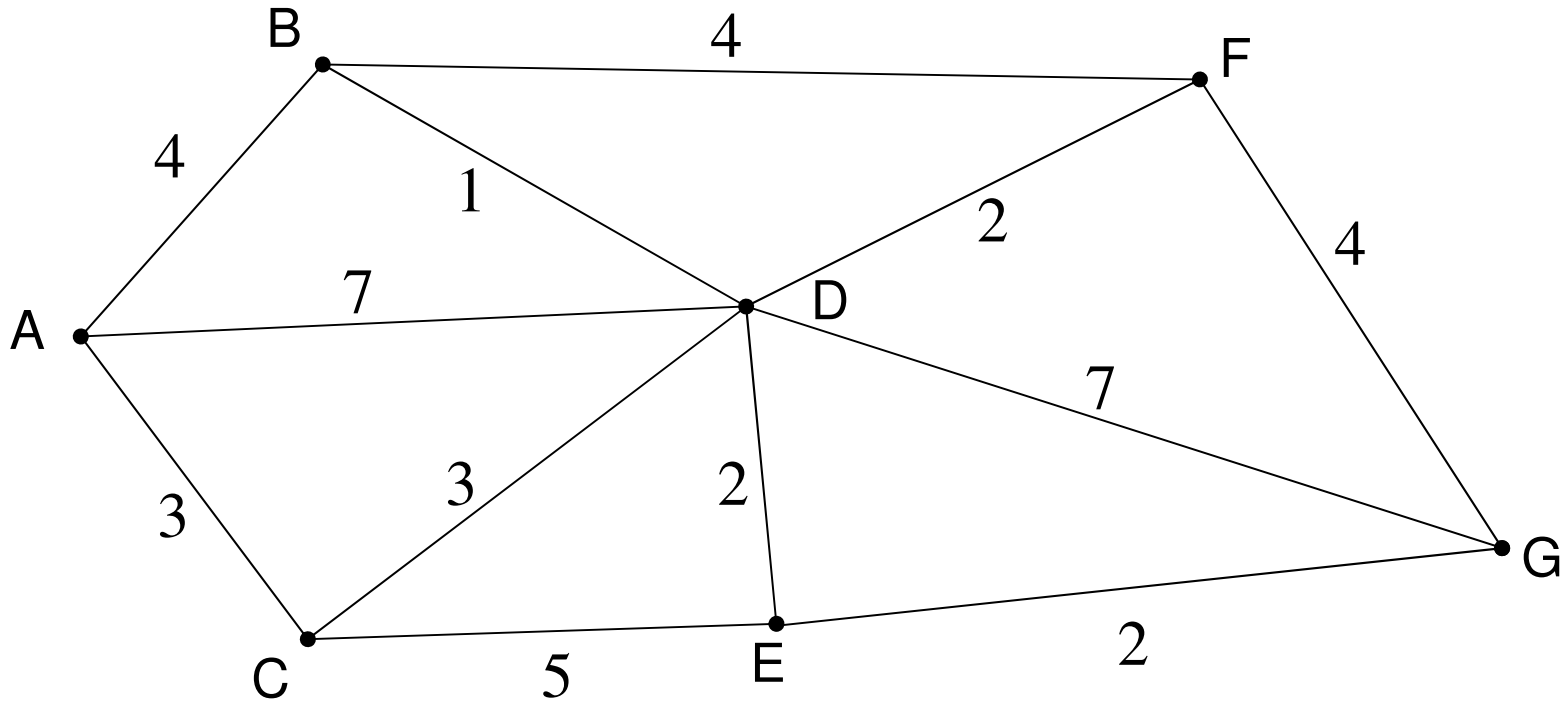


Edsger W. Dijkstra (1930-2002)
One of the most influential figures of computing science's founding generation, Dijkstra was a theoretical physicist whose career was a computer programmer. His ideas lay the foundations for the birth and development of the professional discipline of software engineering. And he gave his name to one of the most famous algorithms in graph theory.

https://en.wikipedia.org/wiki/Edsger_W._Dijkstra

Dijkstra's Algorithm

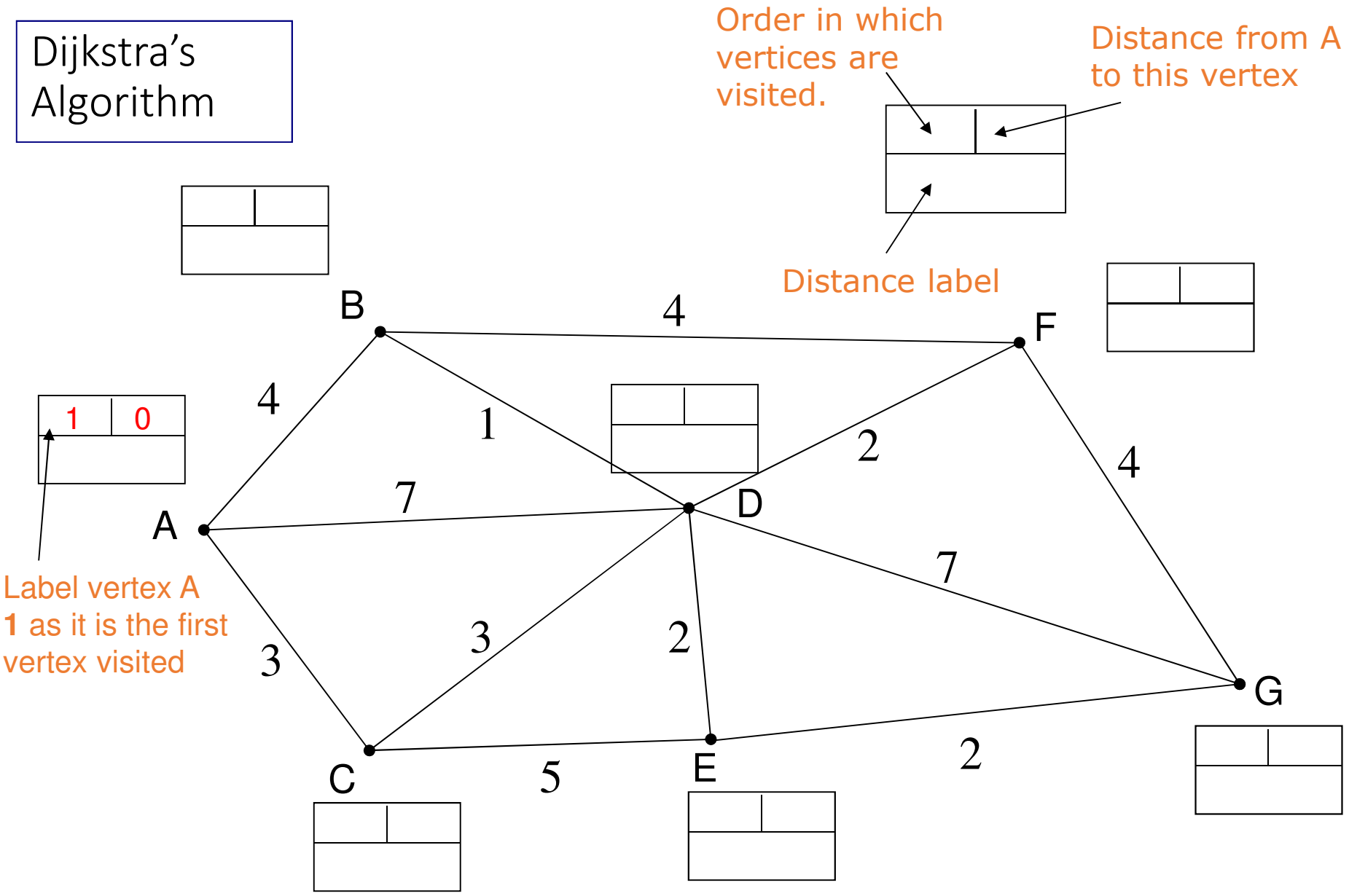
Find the shortest path from a given start vertex to a finishing vertex in the network. We will find the shortest path from A to G by backward reasoning



Dijkstra's Algorithm

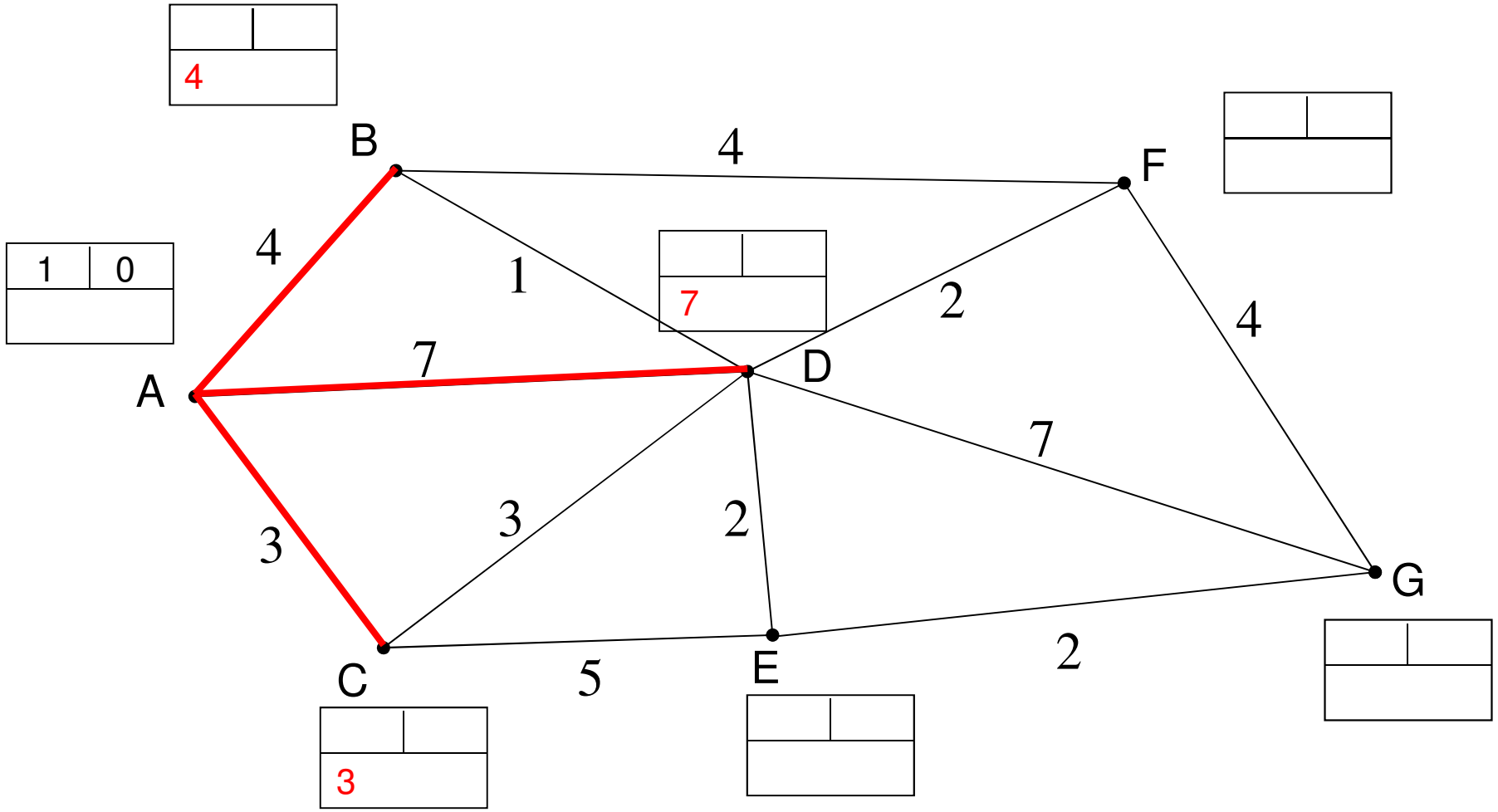
1. Initialize the start vertex with distance label 0 and "visited order" 1
2. Assign temporary distance labels to all the vertices that can be reached directly from the start vertex
3. Select the vertex with the smallest temporary distance label and make this distance label permanent. Update this vertex as "visited" with a "visited order" index incremented by one.
4. Put temporary distance labels on each neighboring one-hop vertex from the vertex you have just made permanent. The temporary distance label is equal to the sum of the permanent distance label plus the connecting edge value. Replace an existing temporary distance label at a vertex only if this new sum is smaller.
5. Go to Step 3.
6. Repeat until the finishing vertex has a permanent label.
7. To find the shortest paths(s), trace back from the end vertex to the start vertex.

Dijkstra's Algorithm

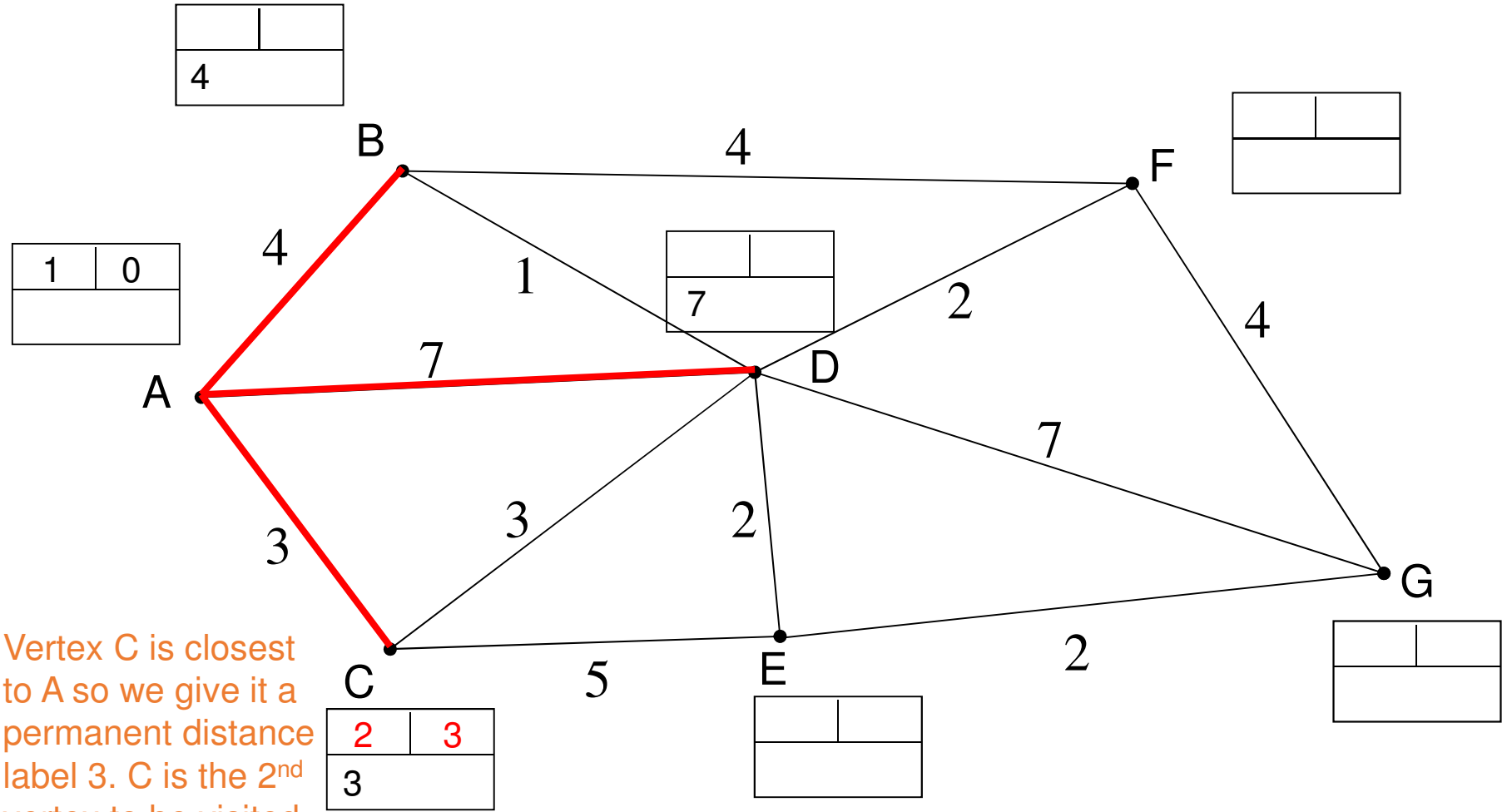


Dijkstra's Algorithm

We update each vertex adjacent to A with a 'working value' for its distance from A.



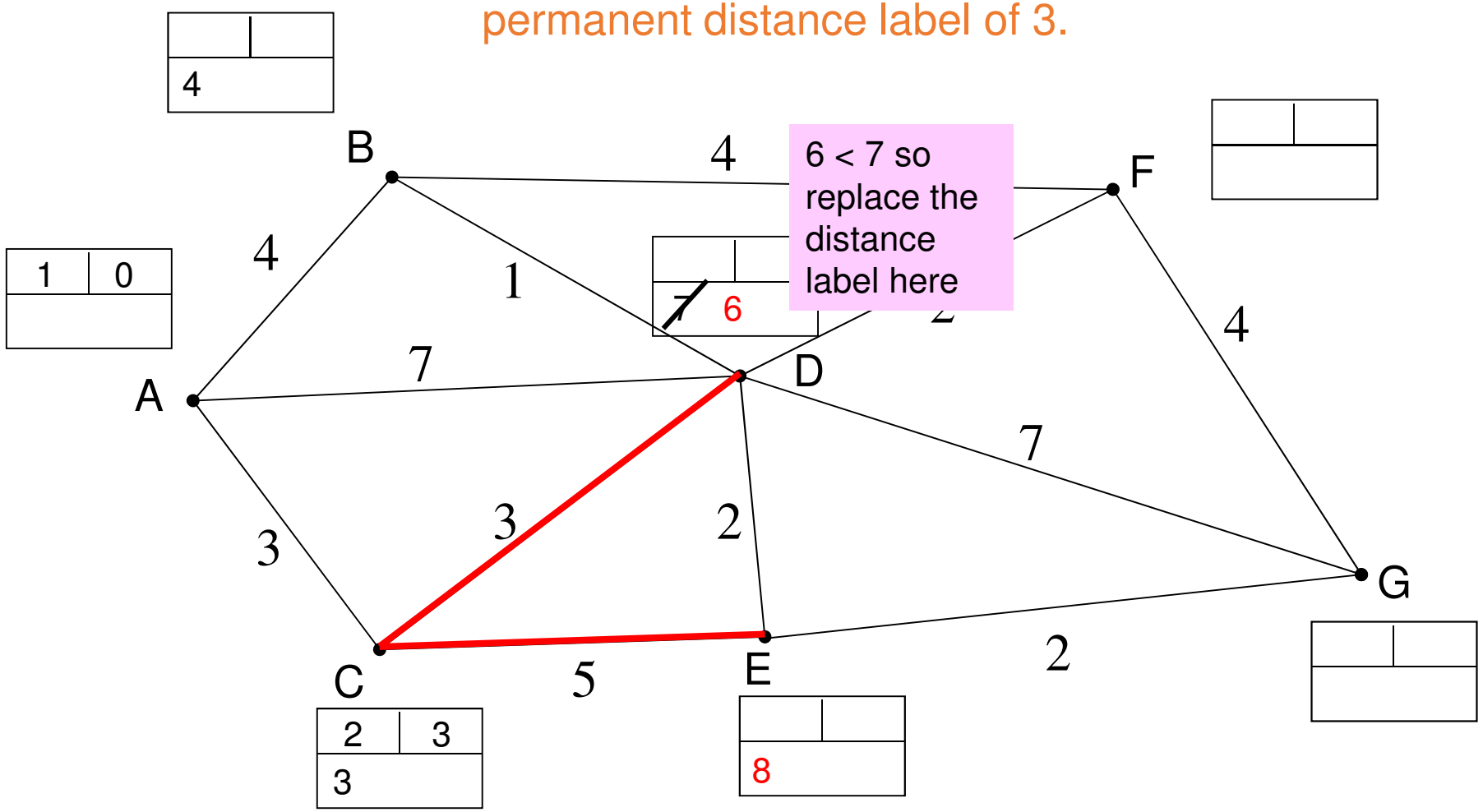
Dijkstra's Algorithm



Vertex C is closest to A so we give it a permanent distance label 3. C is the 2nd vertex to be visited and permanently labelled.

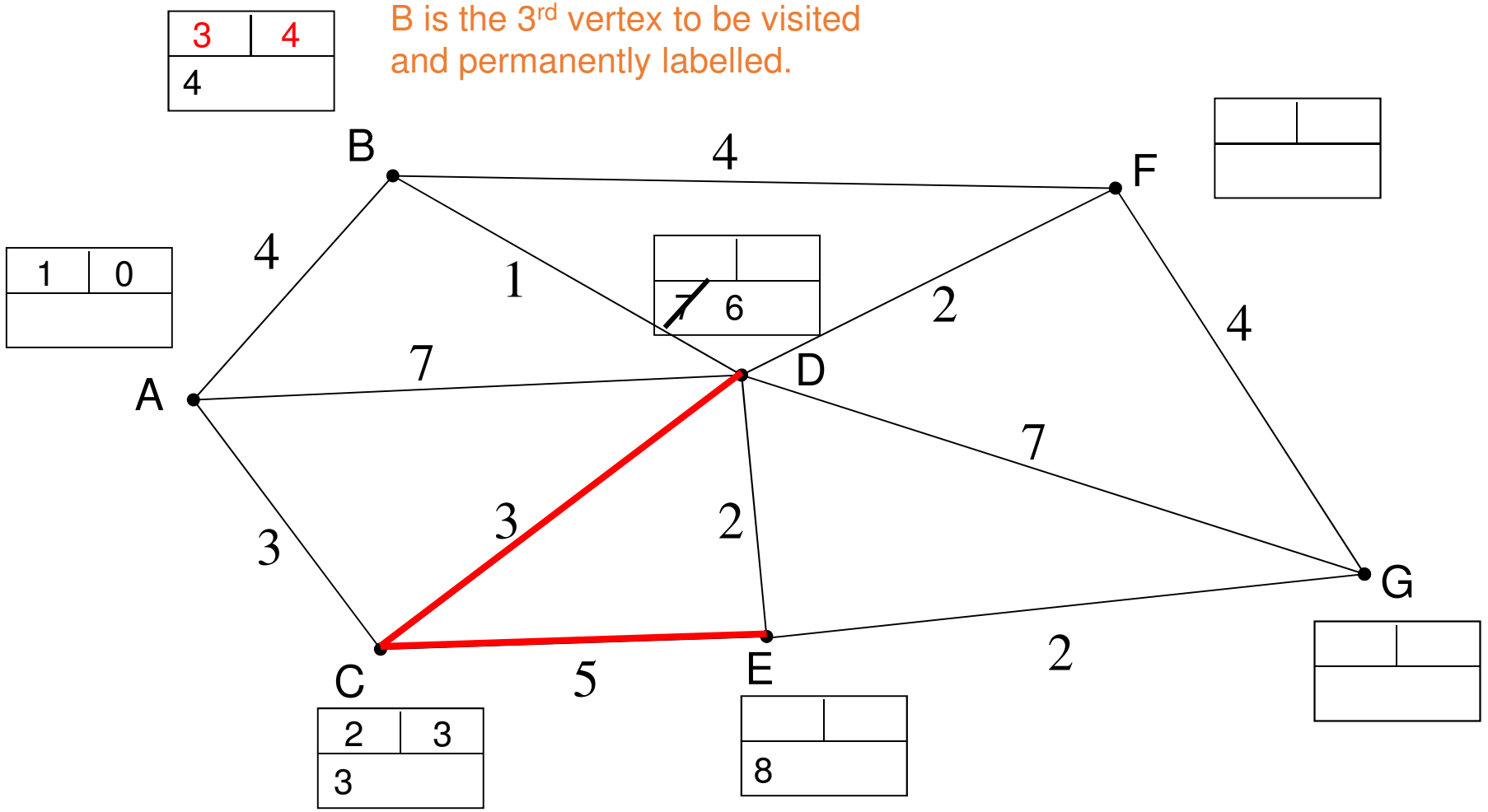
Dijkstra's Algorithm

We update each vertex adjacent to C with a 'working distance value' for its total distance from A, by adding its distance from C to C's permanent distance label of 3.



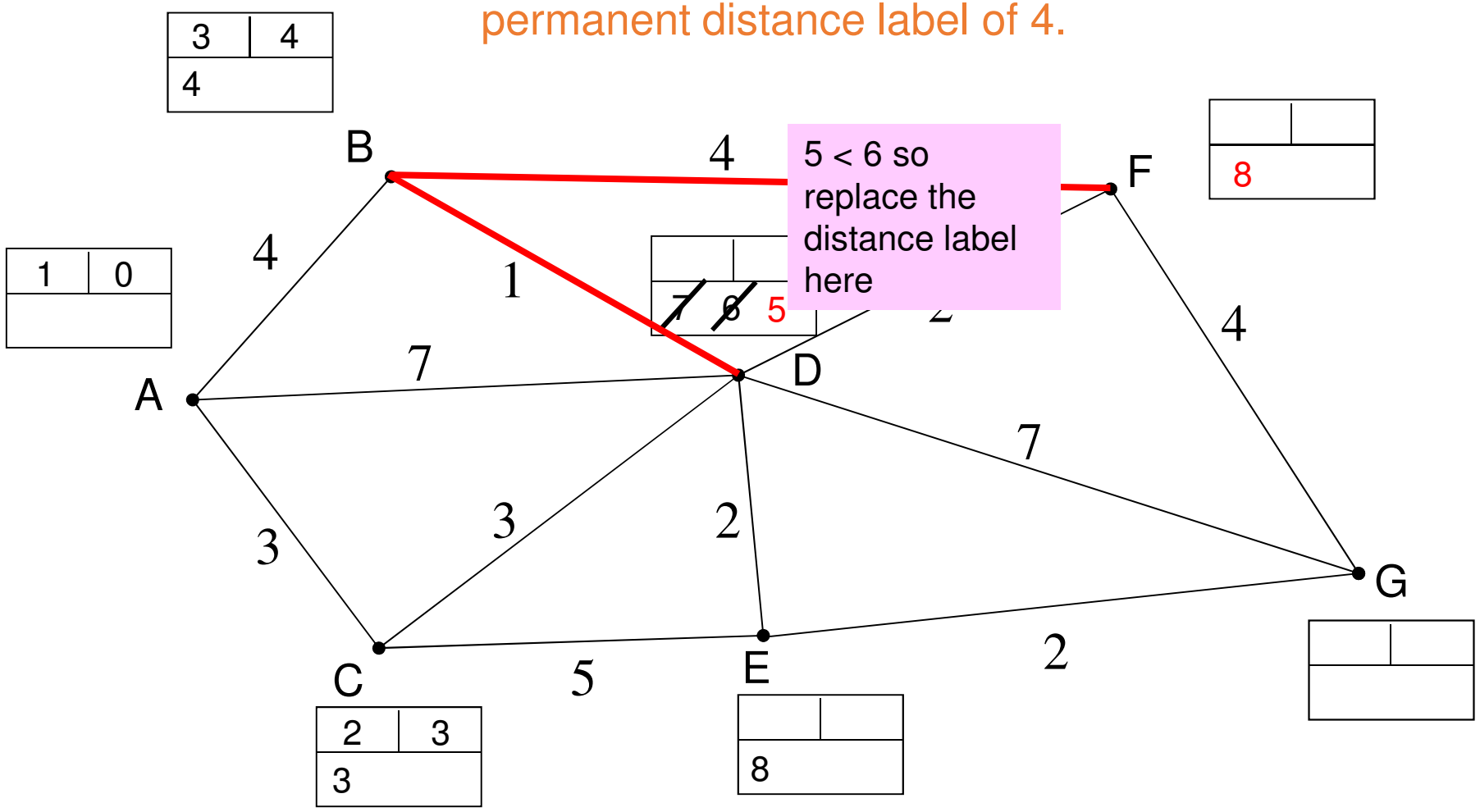
Dijkstra's Algorithm

The vertex with the smallest temporary distance label is B, so make this label permanent. B is the 3rd vertex to be visited and permanently labelled.



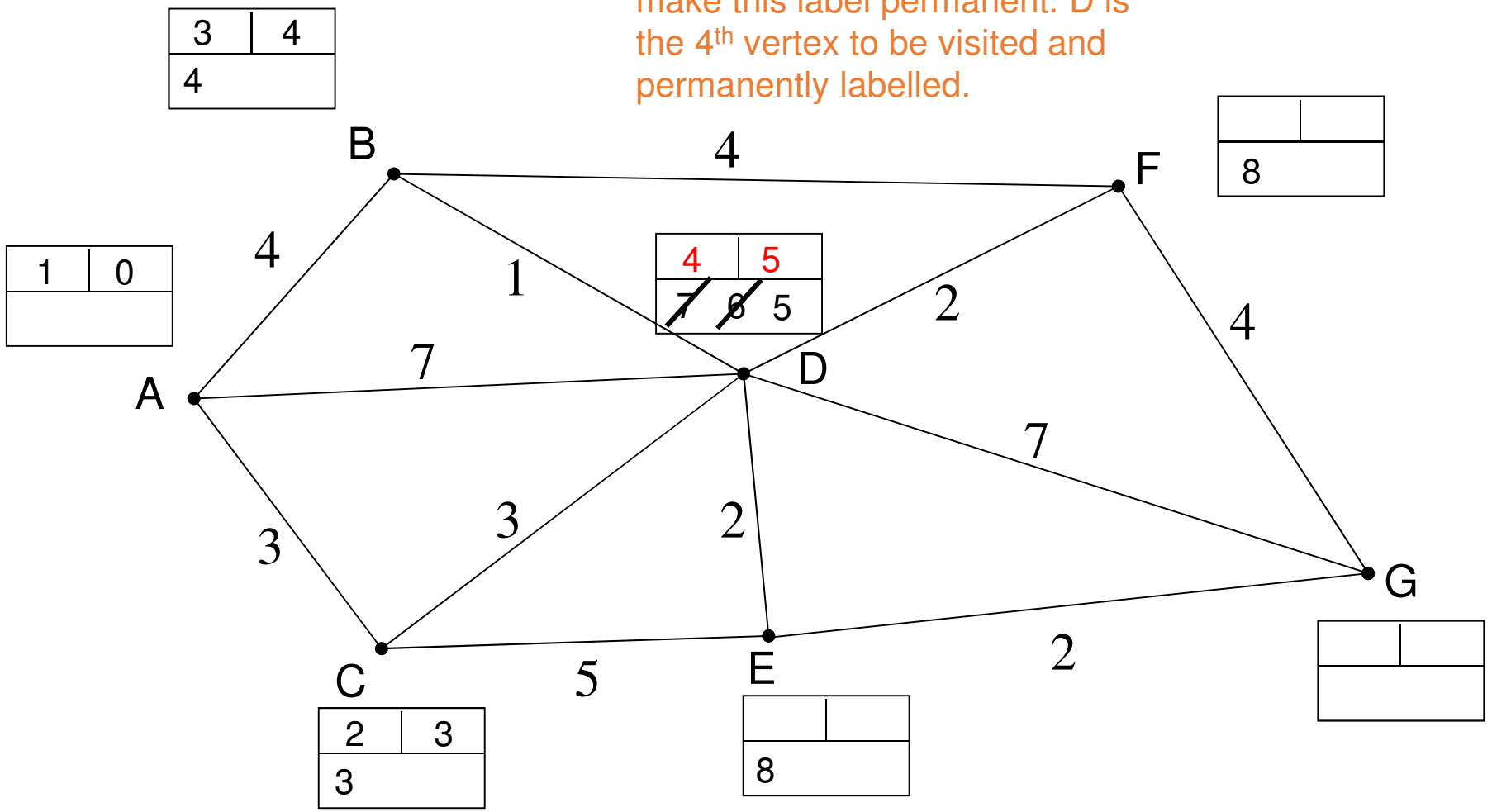
Dijkstra's Algorithm

We update each vertex adjacent to B with a 'working distance value' for its total distance from A, by adding its distance from B to B's permanent distance label of 4.



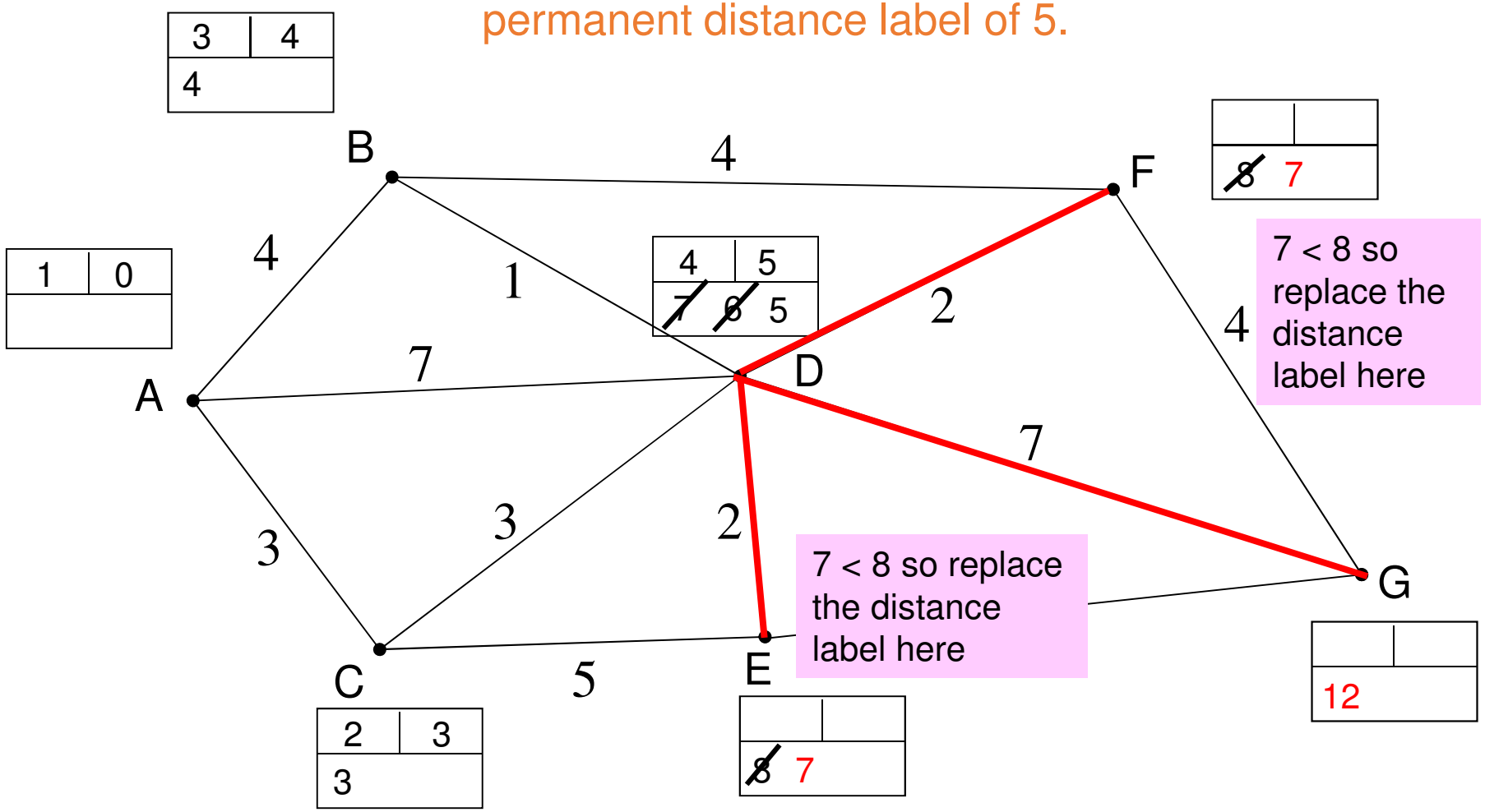
Dijkstra's Algorithm

The vertex with the smallest temporary distance label is D, so make this label permanent. D is the 4th vertex to be visited and permanently labelled.

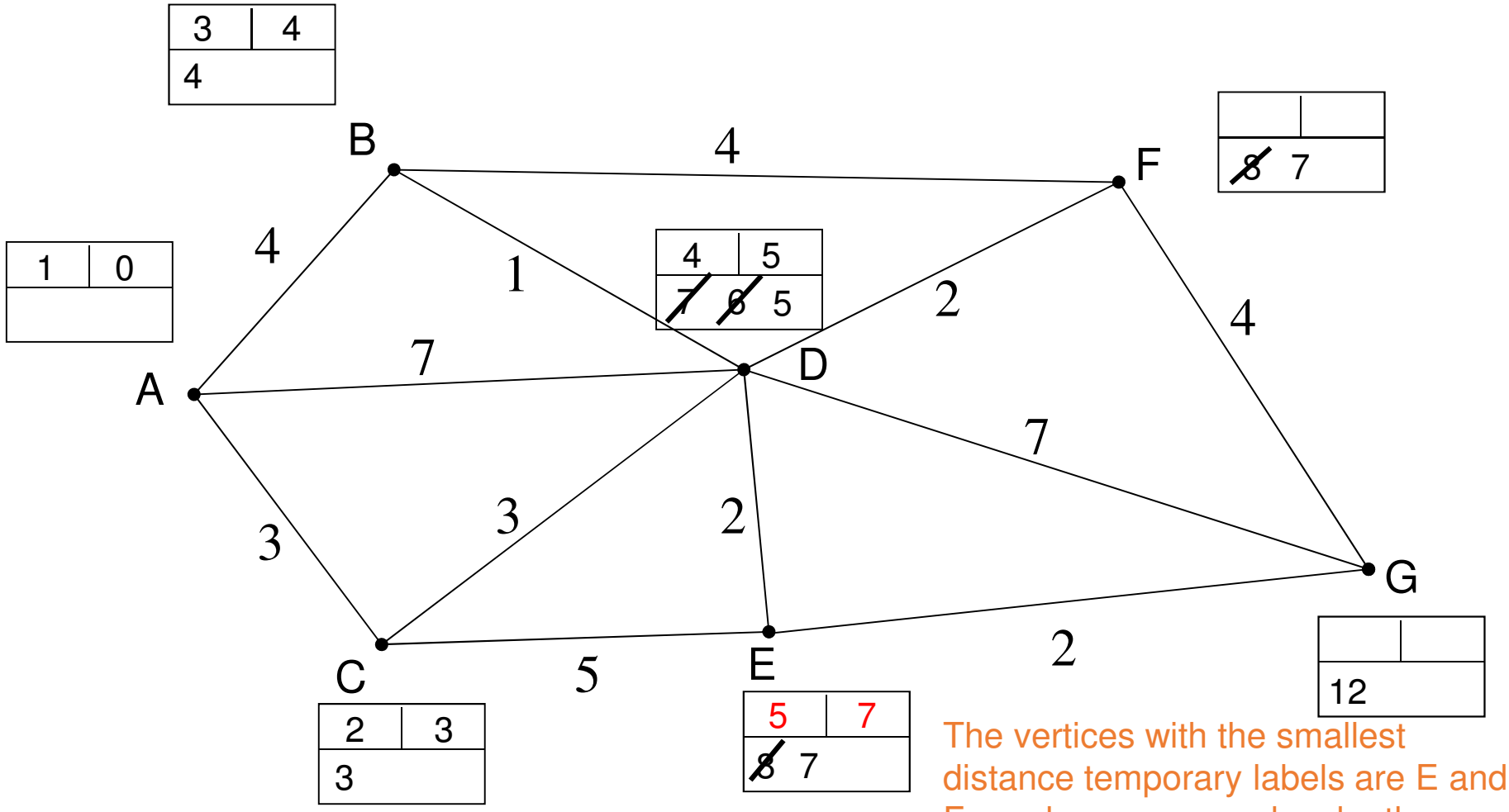


Dijkstra's Algorithm

We update each vertex adjacent to D with a 'working distance value' for its total distance from A, by adding its distance from D to D's permanent distance label of 5.



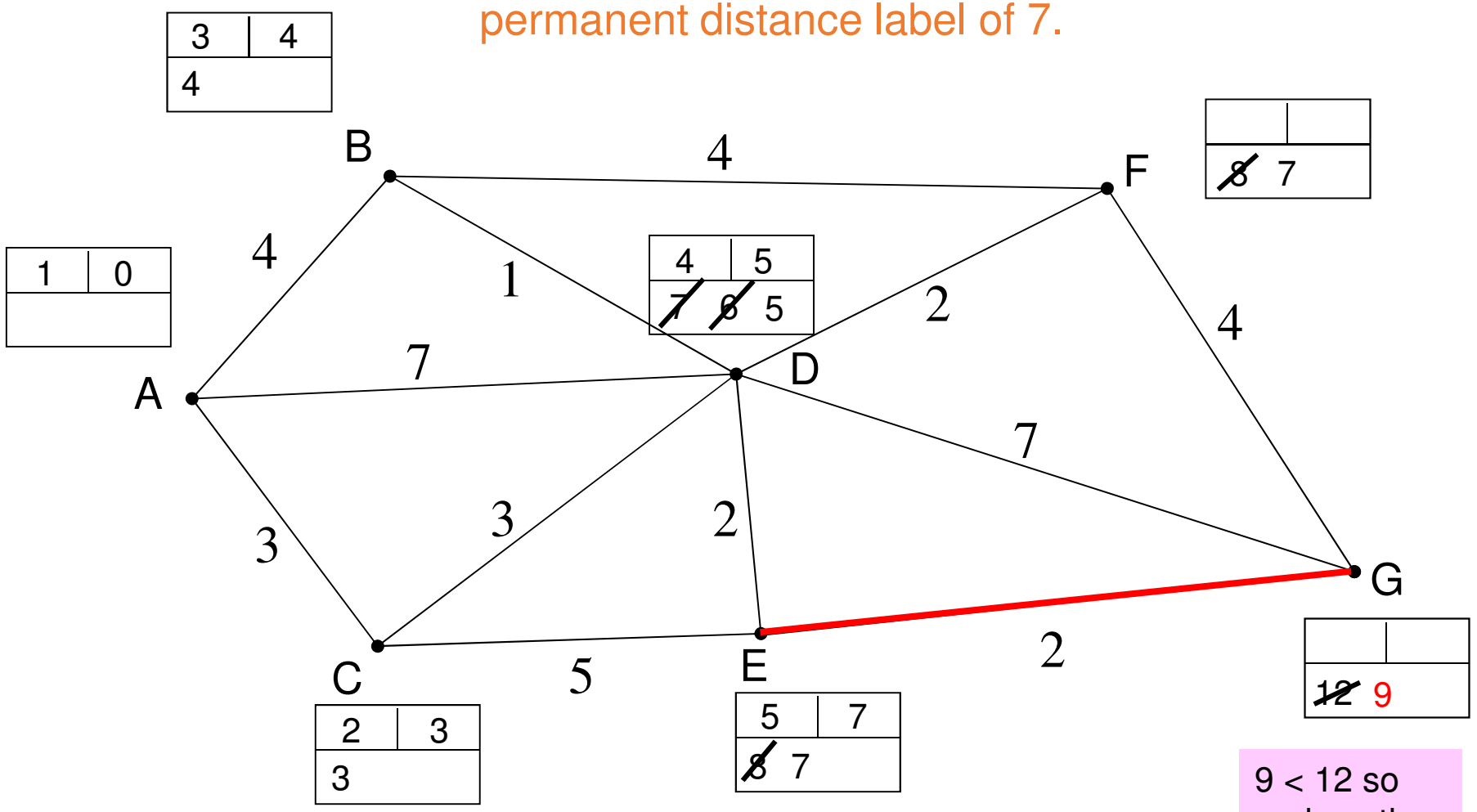
Dijkstra's Algorithm



The vertices with the smallest distance temporary labels are E and F, so choose one and make the label permanent. E is chosen - the 5th vertex to be visited and permanently labelled.

Dijkstra's Algorithm

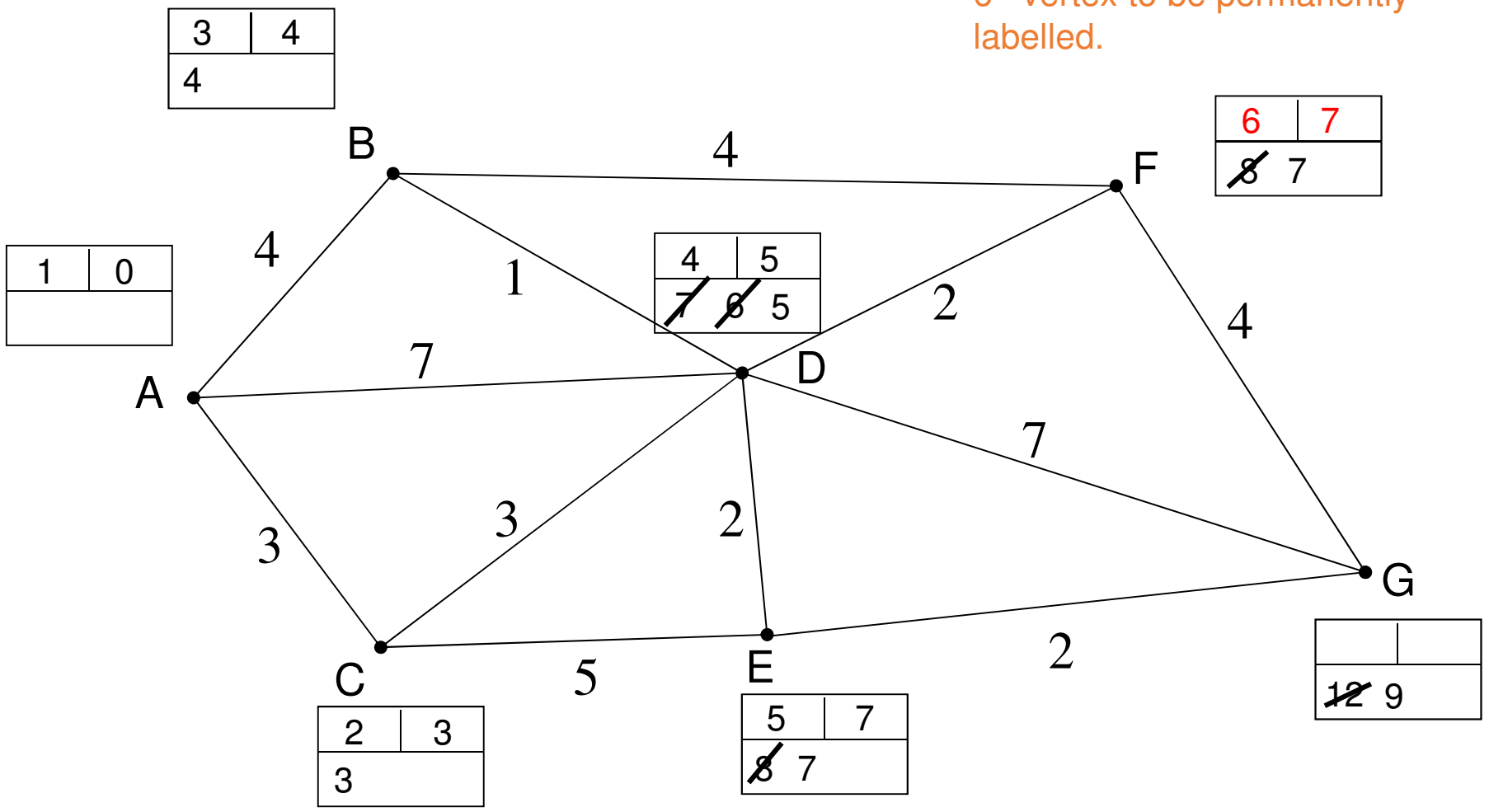
We update each vertex adjacent to E with a 'working distance value' for its total distance from A, by adding its distance from E to E's permanent distance label of 7.



9 < 12 so replace the distance label here

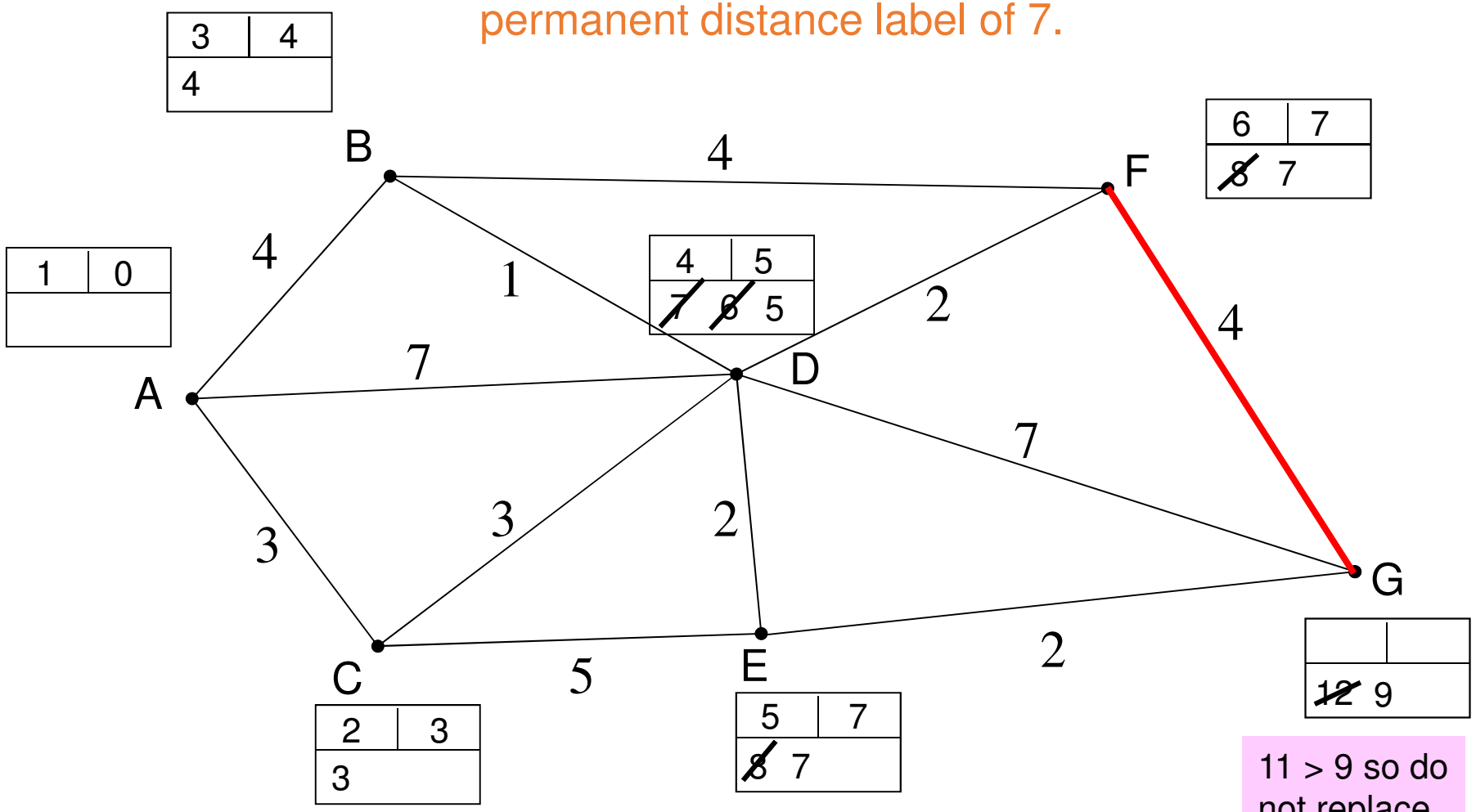
Dijkstra's Algorithm

The vertex with the smallest temporary label is F, so make this label permanent.F is the 6th vertex to be permanently labelled.

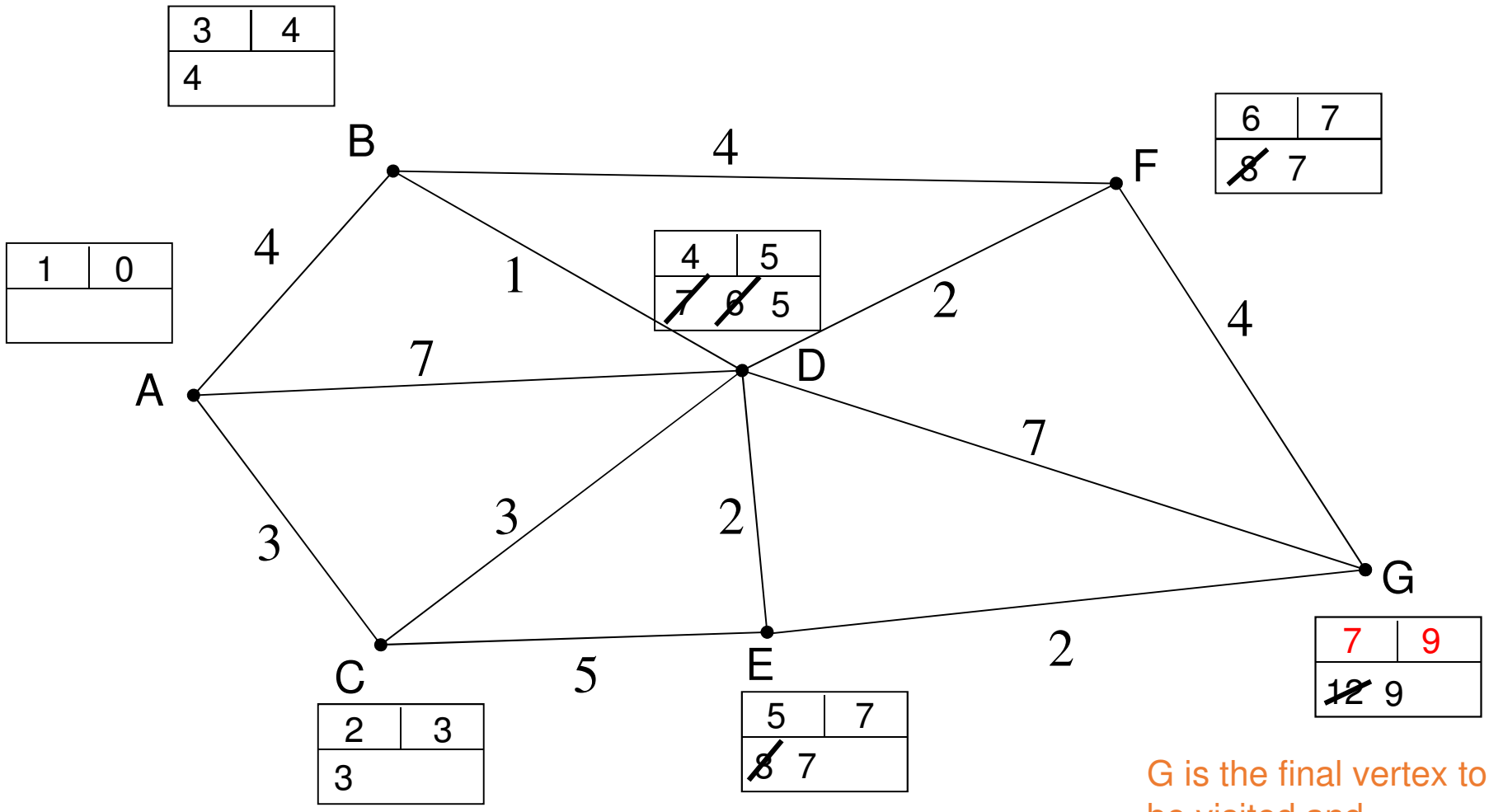


Dijkstra's Algorithm

We update each vertex adjacent to F with a 'working distance value' for its total distance from A, by adding its distance from F to F's permanent distance label of 7.



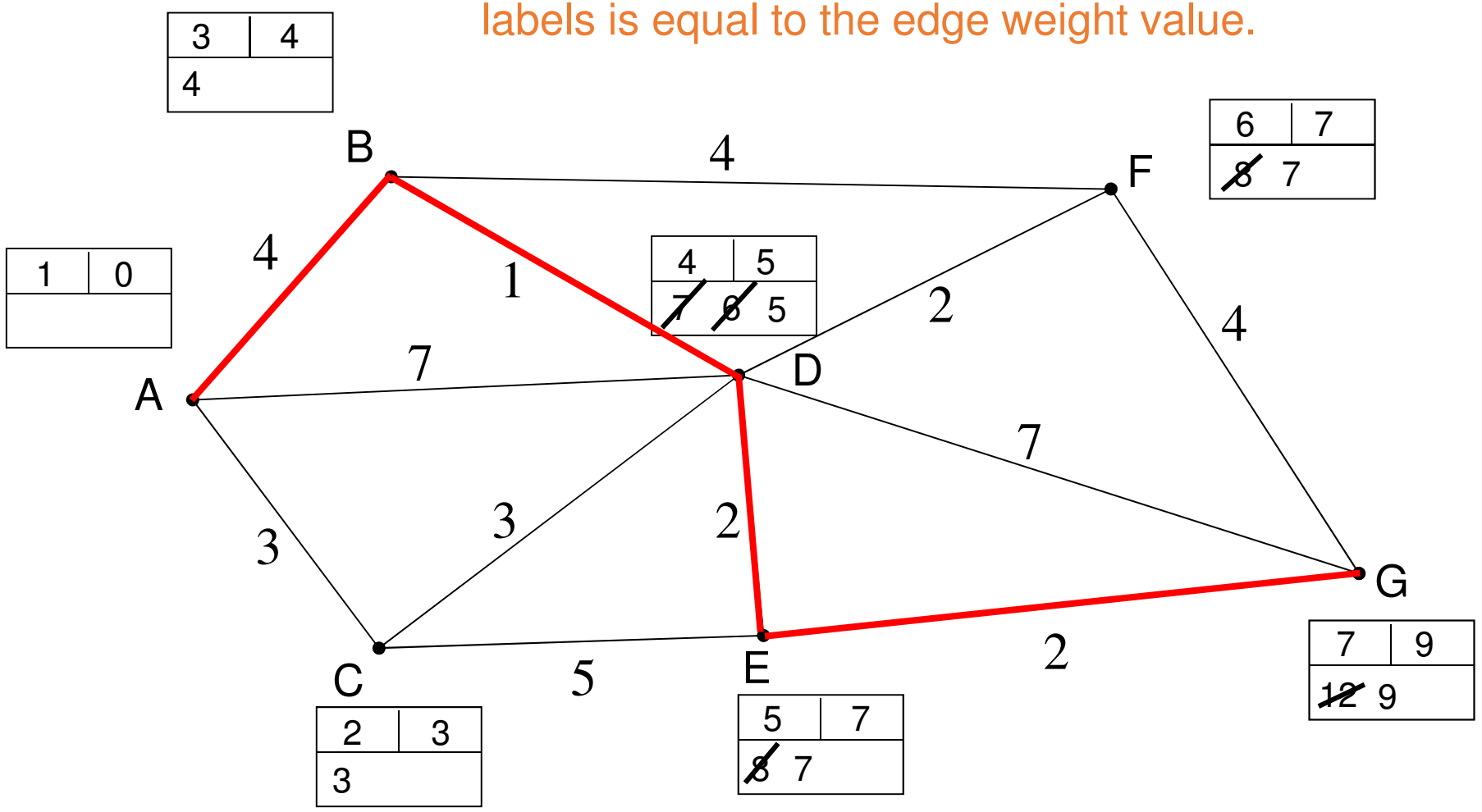
Dijkstra's Algorithm



G is the final vertex to be visited and permanently labelled.

Dijkstra's Algorithm

To find the shortest path from A to G, start from G and work backwards, choosing edges for which the difference between the permanent distance labels is equal to the edge weight value.



The shortest path is ABDEG, with length 9.