C: Chaotic Crossing

Problem author: Reinier Schmiermann

Problem: Given an undirected graph. Each edge ij has a probability p_{ij} that it can be used during each second, and crossing an edge takes exactly one second. What is the minimum expected time to go from s to t?

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Problem: Given an undirected graph. Each edge ij has a probability p_{ij} that it can be used during each second, and crossing an edge takes exactly one second. What is the minimum expected time to go from s to t?

Strategy: Each vertex i has some minimum expected time E_i to go from i to B. For t, we have $E_t = 0$. For any other vertex i, each second, we only cross an edge if we can move to a vertex j with $E_j < E_i$, and if we can, then we pick the smallest E_j.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Problem: Given an undirected graph. Each edge ij has a probability p_{ij} that it can be used during each second, and crossing an edge takes exactly one second. What is the minimum expected time to go from s to t?

Strategy: Each vertex i has some minimum expected time E_i to go from i to B. For t, we have $E_t = 0$. For any other vertex i, each second, we only cross an edge if we can move to a vertex j with $E_j < E_i$, and if we can, then we pick the smallest E_j.

Algebra: For example, if vertex 4 neighbors vertices $1, 2, 3$ with $E_1 < E_2 < E_3 < E_4$, we get

$$E_4 = 1 + p_{41}E_1 + (1 - p_{41})p_{42}E_2 + (1 - p_{41})(1 - p_{42})p_{43}E_3 + (1 - p_{41})(1 - p_{42})(1 - p_{43})E_4.$$

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Problem: Given an undirected graph. Each edge ij has a probability p_{ij} that it can be used during each second, and crossing an edge takes exactly one second. What is the minimum expected time to go from s to t?

Strategy: Each vertex i has some minimum expected time E_i to go from i to B. For t, we have $E_t = 0$. For any other vertex i, each second, we only cross an edge if we can move to a vertex j with $E_j < E_i$, and if we can, then we pick the smallest E_j.

Algebra: For example, if vertex 4 neighbors vertices $1, 2, 3$ with $E_1 < E_2 < E_3 < E_4$, we get

$$E_4 = 1 + p_{41}E_1 + (1 - p_{41})p_{42}E_2 + (1 - p_{41})(1 - p_{42})p_{43}E_3 + (1 - p_{41})(1 - p_{42})(1 - p_{43})E_4.$$

If we write this as $E_4 = A_4 + B_4E_4$, then this solves to $E_4 = \frac{A_4}{1 - B_4}$.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Problem: Given an undirected graph. Each edge ij has a probability p_{ij} that it can be used during each second, and crossing an edge takes exactly one second. What is the minimum expected time to go from s to t?

Strategy: Each vertex i has some minimum expected time E_i to go from i to B. For t, we have $E_t = 0$. For any other vertex i, each second, we only cross an edge if we can move to a vertex j with $E_j < E_i$, and if we can, then we pick the smallest E_j.

Algebra: For example, if vertex 4 neighbors vertices $1, 2, 3$ with $E_1 < E_2 < E_3 < E_4$, we get

$$E_4 = 1 + p_{41}E_1 + (1 - p_{41})p_{42}E_2 + (1 - p_{41})(1 - p_{42})p_{43}E_3 + (1 - p_{41})(1 - p_{42})(1 - p_{43})E_4.$$

If we write this as $E_4 = A_4 + B_4E_4$, then this solves to $E_4 = \frac{A_4}{1 - B_4}$.

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_ip_{ij}E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1 - B_i}$. If $B_i = 1$, then t cannot be reached.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

Algorithm: Initialize $A_i = 1$ and $B_i = 1$ for each $i \neq t$. Keep a priority queue of possible valuations of the values E_i, starting with $E_t = 0$.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

Algorithm: Initialize $A_i = 1$ and $B_i = 1$ for each $i \neq t$. Keep a priority queue of possible valuations of the values E_i, starting with $E_t = 0$.

Repeatedly take the lowest possible valuation $E_i = x$ from the heap and assign the value if E_i wasn't already assigned. Update all neighbors j, increasing A_j by $B_j p_{ij} E_i$ and multiplying B_j by $1 - p_{ij}$, and adding the possible valuation $E_j = \frac{A_j}{1-B_j}$.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

Algorithm: Initialize $A_i = 1$ and $B_i = 1$ for each $i \neq t$. Keep a priority queue of possible valuations of the values E_i, starting with $E_t = 0$.

Repeatedly take the lowest possible valuation $E_i = x$ from the heap and assign the value if E_i wasn't already assigned. Update all neighbors j, increasing A_j by $B_j p_{ij} E_i$ and multiplying B_j by $1 - p_{ij}$, and adding the possible valuation $E_j = \frac{A_j}{1-B_j}$.

When the queue is empty, we either found E_s, or there is no path from s to t.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

Algorithm: Initialize $A_i = 1$ and $B_i = 1$ for each $i \neq t$. Keep a priority queue of possible valuations of the values E_i, starting with $E_t = 0$.

Repeatedly take the lowest possible valuation $E_i = x$ from the heap and assign the value if E_i wasn't already assigned. Update all neighbors j, increasing A_j by $B_j p_{ij} E_i$ and multiplying B_j by $1 - p_{ij}$, and adding the possible valuation $E_j = \frac{A_j}{1-B_j}$.

When the queue is empty, we either found E_s, or there is no path from s to t.

Running time: $\mathcal{O}((n + m)\log(n + m))$ with a binary heap.

C: Chaotic Crossing

Problem author: Reinier Schmiermann

Calculation: To calculate A_i and B_i, we initialize $A_i = 1$ and $B_i = 1$. We then iterate the neighbors j with $E_j < E_i$ in increasing order of E_j, and we increase A_i by $B_i p_{ij} E_j$ and multiply B_i by $1 - p_{ij}$. Then $E_i = \frac{A_i}{1-B_i}$. If $B_i = 1$, then t cannot be reached.

Issue: The values of E_i depend on eachother, so where do we start?

Observation: The value E_i only depends on the values E_j of vertices j with $E_j < E_i$. So we can start at $E_t = 0$, and determine the rest of the values E_i in increasing order of E_i.

Algorithm: Initialize $A_i = 1$ and $B_i = 1$ for each $i \neq t$. Keep a priority queue of possible valuations of the values E_i, starting with $E_t = 0$.

Repeatedly take the lowest possible valuation $E_i = x$ from the heap and assign the value if E_i wasn't already assigned. Update all neighbors j, increasing A_j by $B_j p_{ij} E_i$ and multiplying B_j by $1 - p_{ij}$, and adding the possible valuation $E_j = \frac{A_j}{1-B_j}$.

When the queue is empty, we either found E_s, or there is no path from s to t.

Running time: $\mathcal{O}((n+m)\log(n+m))$ with a binary heap.

Statistics: 7 submissions, 1 accepted, 5 unknown