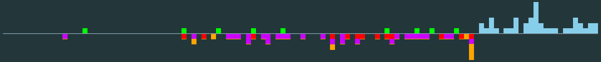
D: Duo Detection

Problem author: Mike de Vries

Problem: Given n messages, M_i, find two messages which have at least 2 symbols in common. The total size of all messages is not more than $m = 100000$.

D: Duo Detection

Problem author: Mike de Vries

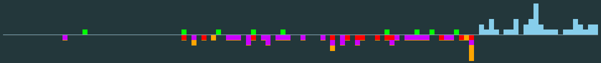


Problem: Given n messages, M_i, find two messages which have at least 2 symbols in common. The total size of all messages is not more than $m = 100000$.

Observation: Build a bipartite graph with on one side the messages, and on the other the symbols. Want to find a 4-cycle in the graph.

D: Duo Detection

Problem author: Mike de Vries



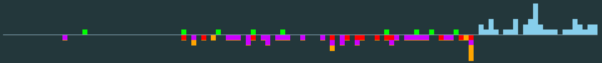
Problem: Given n messages, M_i, find two messages which have at least 2 symbols in common. The total size of all messages is not more than $m = 100000$.

Observation: Build a bipartite graph with on one side the messages, and on the other the symbols. Want to find a 4-cycle in the graph.

Naive solution 1: Loop over all pairs of messages, and calculate the intersection of their symbol sets in $\mathcal{O}(\min(|M_i|, |M_j|))$, using hash sets or boolean arrays after using coordinate compression. Time complexity, roughly: $\mathcal{O}(n \sum_{i=1}^{n} |M_i|)$

D: Duo Detection

Problem author: Mike de Vries



Problem: Given n messages, M_i, find two messages which have at least 2 symbols in common. The total size of all messages is not more than $m = 100000$.

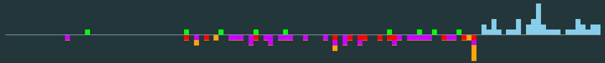
Observation: Build a bipartite graph with on one side the messages, and on the other the symbols. Want to find a 4-cycle in the graph.

Naive solution 1: Loop over all pairs of messages, and calculate the intersection of their symbol sets in $\mathcal{O}(\min(|M_i|, |M_j|))$, using hash sets or boolean arrays after using coordinate compression. Time complexity, roughly: $\mathcal{O}(n \sum_{i=1}^{n} |M_i|)$

When n is small, works well, but can be quadratic.

D: Duo Detection

Problem author: Mike de Vries



Problem: Given n messages, M_i, find two messages which have at least 2 symbols in common. The total size of all messages is not more than $m = 100000$.

Observation: Build a bipartite graph with on one side the messages, and on the other the symbols. Want to find a 4-cycle in the graph.

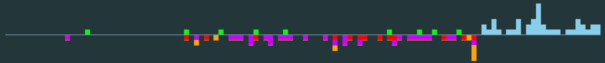
Naive solution 1: Loop over all pairs of messages, and calculate the intersection of their symbol sets in $\mathcal{O}(\min(|M_i|, |M_j|))$, using hash sets or boolean arrays after using coordinate compression. Time complexity, roughly: $\mathcal{O}(n \sum_{i=1}^{n} |M_i|)$

When n is small, works well, but can be quadratic.

Naive solution 2: Loop over all pairs of symbols in each message. Check if any of these pairs is the same, by using a hash map. Time complexity: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$ Works well when the sizes of the messages are small.

D: Duo Detection

Problem author: Mike de Vries



Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

D: Duo Detection

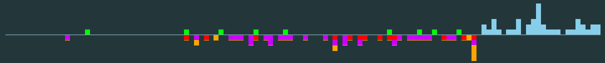
Problem author: Mike de Vries

Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

Solution: The time limit and constraints are generous. Combine the naive solutions in a smart way to obtain a faster algorithm in the worst case.
Divide messages in big and small messages, based on parameter B. Do casework:

D: Duo Detection

Problem author: Mike de Vries



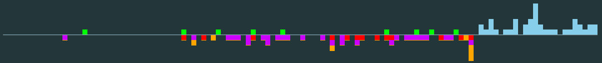
Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

Solution: The time limit and constraints are generous. Combine the naive solutions in a smart way to obtain a faster algorithm in the worst case.
Divide messages in big and small messages, based on parameter B. Do casework:

Big-Big Use naive solution 1. Works in $\mathcal{O}(\sum_{i \in \text{Bigs}} |M_i| \frac{m}{B}) = \mathcal{O}(\frac{m^2}{B})$

D: Duo Detection

Problem author: Mike de Vries



Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

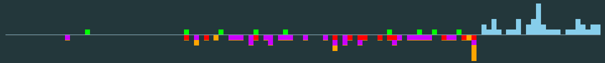
Solution: The time limit and constraints are generous. Combine the naive solutions in a smart way to obtain a faster algorithm in the worst case.
Divide messages in big and small messages, based on parameter B. Do casework:

Big-Big Use naive solution 1. Works in $\mathcal{O}(\sum_{i \in \text{Bigs}} |M_i| \frac{m}{B}) = \mathcal{O}(\frac{m^2}{B})$

Small-Big Use naive solution 1. Because of the minimum over the two message lengths, still $\mathcal{O}(\frac{m^2}{B})$

D: Duo Detection

Problem author: Mike de Vries



Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

Solution: The time limit and constraints are generous. Combine the naive solutions in a smart way to obtain a faster algorithm in the worst case.
Divide messages in big and small messages, based on parameter B. Do casework:

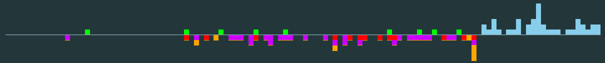
Big-Big Use naive solution 1. Works in $\mathcal{O}(\sum_{i \in \text{Bigs}} |M_i| \frac{m}{B}) = \mathcal{O}(\frac{m^2}{B})$

Small-Big Use naive solution 1. Because of the minimum over the two message lengths, still $\mathcal{O}(\frac{m^2}{B})$

Small-Small Use naive solution 2, works in $\mathcal{O}(mB)$ (worst case is all small messages are equal in size and $< B$.)
This handles all the cases, and total complexity is $\mathcal{O}(\frac{m^2}{B} + mB)$, best when $B = \Theta(\sqrt{m})$.

D: Duo Detection

Problem author: Mike de Vries



Reminder: Naive 1: $\mathcal{O}(\min(|M_i|, |M_j|))$ over all pairs of messages.
Naive 2: $\mathcal{O}(\sum_{i=1}^{n} |M_i|^2)$

Solution: The time limit and constraints are generous. Combine the naive solutions in a smart way to obtain a faster algorithm in the worst case.
Divide messages in big and small messages, based on parameter B. Do casework:

Big-Big Use naive solution 1. Works in $\mathcal{O}(\sum_{i \in \text{Bigs}} |M_i| \frac{m}{B}) = \mathcal{O}(\frac{m^2}{B})$

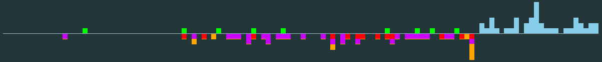
Small-Big Use naive solution 1. Because of the minimum over the two message lengths, still $\mathcal{O}(\frac{m^2}{B})$

Small-Small Use naive solution 2, works in $\mathcal{O}(mB)$ (worst case is all small messages are equal in size and $< B$.)
This handles all the cases, and total complexity is $\mathcal{O}(\frac{m^2}{B} + mB)$, best when $B = \Theta(\sqrt{m})$.

Running time: $\mathcal{O}(m\sqrt{m})$ with a hash map. The algorithm has a considerable constant factor.

D: Duo Detection

Problem author: Mike de Vries



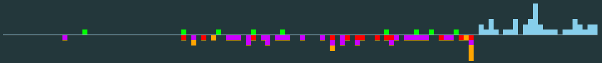
Bonus: You can get rid of the hash map. This requires in naive solution 2 to order the computations in a smart way, such that a global boolean array can be used, which is set and unset for each message. Also requires sorting and coordinate compression beforehand. Same tricks need to be used in naive solution 1.

Bonus: You can get rid of the hash map. This requires in naive solution 2 to order the computations in a smart way, such that a global boolean array can be used, which is set and unset for each message. Also requires sorting and coordinate compression beforehand. Same tricks need to be used in naive solution 1.

Bonus 2: The algorithm can be sped up to $\mathcal{O}(m\sqrt{\frac{m}{w}})$, where w is the word size of the machine (typically 32 or 64). This can be done with the use of bitsets in naive algorithm 1. The details are an exercise for the interested reader.

D: Duo Detection

Problem author: Mike de Vries



Bonus: You can get rid of the hash map. This requires in naive solution 2 to order the computations in a smart way, such that a global boolean array can be used, which is set and unset for each message. Also requires sorting and coordinate compression beforehand. Same tricks need to be used in naive solution 1.

Bonus 2: The algorithm can be sped up to $\mathcal{O}(m\sqrt{\frac{m}{w}})$, where w is the word size of the machine (typically 32 or 64). This can be done with the use of bitsets in naive algorithm 1. The details are an exercise for the interested reader.

Alternative solution: A time complexity of $\mathcal{O}(\frac{m^2}{w})$ can also get accepted when implemented well. (This is basically doing bonus 2 without the square-root trick).

Bonus: You can get rid of the hash map. This requires in naive solution 2 to order the computations in a smart way, such that a global boolean array can be used, which is set and unset for each message. Also requires sorting and coordinate compression beforehand. Same tricks need to be used in naive solution 1.

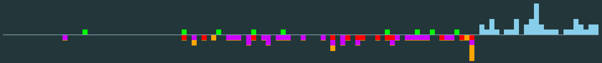
Bonus 2: The algorithm can be sped up to $\mathcal{O}(m\sqrt{\frac{m}{w}})$, where w is the word size of the machine (typically 32 or 64). This can be done with the use of bitsets in naive algorithm 1. The details are an exercise for the interested reader.

Alternative solution: A time complexity of $\mathcal{O}(\frac{m^2}{w})$ can also get accepted when implemented well. (This is basically doing bonus 2 without the square-root trick).

Fun fact: Constructions similar to problem F (Faulty Connection) are used in the testdata to make dense testcases with a impossible answer.

D: Duo Detection

Problem author: Mike de Vries



Bonus: You can get rid of the hash map. This requires in naive solution 2 to order the computations in a smart way, such that a global boolean array can be used, which is set and unset for each message. Also requires sorting and coordinate compression beforehand. Same tricks need to be used in naive solution 1.

Bonus 2: The algorithm can be sped up to $\mathcal{O}(m\sqrt{\frac{m}{w}})$, where w is the word size of the machine (typically 32 or 64). This can be done with the use of bitsets in naive algorithm 1. The details are an exercise for the interested reader.

Alternative solution: A time complexity of $\mathcal{O}(\frac{m^2}{w})$ can also get accepted when implemented well. (This is basically doing bonus 2 without the square-root trick).

Fun fact: Constructions similar to problem F (Faulty Connection) are used in the testdata to make dense testcases with a impossible answer.

Statistics: 98 submissions, 9 accepted, 40 unknown