Week 1 Quiz

What is the complexity of heap sort?

a. $O(n)$
b. $O(n^2)$
c. $O(n \log n)$
d. $O(n^2 \log n)$

Week 1 Quiz

What is the complexity of heap sort?

a. $O(n)$
b. $O(n^2)$
c. $O(n \log n)$
d. $O(n^2 \log n)$

Heap can be built in $O(n)$ and then $O(n \log n)$ to remove maximal elements from the heap and preserve the heap structure n times

Week 1 Quiz

If you perform Radix sort on a string of length `m` using the factor `b` of `m`.

Which of the following statements is true (there is only one correct statement)?

a. The number of iterations is m/b and number of buckets is $2b$
b. The number of iterations is m/b and number of buckets is b^2
c. The number of iterations is b/m and number of buckets is $2b$
d. The number of iterations is b/m and number of buckets is b^2

Week 1 Quiz

If you perform Radix sort on a string of length m using the factor b of m.

Which of the following statements is true (there is only one correct statement)?

a. The number of iterations is m/b and number of buckets is 2b
b. **The number of iterations is m/b and number of buckets is 2^b**
c. The number of iterations is b/m and number of buckets is 2b
d. The number of iterations is b/m and number of buckets is b^2

Split string of length m into m/b blocks of size b
and need a bucket for each bit string of length b

Week 1 Quiz

As we have seen, every comparison based sorting algorithm must use at least `O(n log n)` comparisons.

This is because the decision tree for the algorithm must have at least:

a. n^2 leaf nodes
b. $2n$ branch nodes
c. $n!$ leaf nodes
d. $n!$ branch nodes

Week 1 Quiz

As we have seen, every comparison based sorting algorithm must use at least `O(n log n)` comparisons.

This is because the decision tree for the algorithm must have at least:

a. n^2 leaf nodes
b. 2n branch nodes
c. **n! leaf nodes**
d. n! branch nodes

Leaf nodes are the different possible orderings and n! possibilities

Week 2 Quiz

What is the complexity of breath first search when using an adjacency list representation for an undirected graph with n vertices and m edges?

a. O(n+m)
b. O(log n + log m)
c. O(n^2)
d. O(nm)

Week 2 Quiz

What is the complexity of breath first search when using an adjacency list representation for an undirected graph with n vertices and m edges?

a. **O(n+m)**
b. O(log n + log m)
c. O(n^2)
d. O(nm)

Go through the adjacency list of each vertex once and each edge appears in each edge twice.

Week 2 Quiz

If all edges have the same weight in an undirected graph, which algorithm will find the shortest path between two nodes more efficiently?

a. Dijkstra
b. Bellman-Ford
c. Depth-First Search
d. Breadth-First Search

Week 2 Quiz

If all edges have the same weight in an undirected graph, which algorithm will find the shortest path between two nodes more efficiently?

a. Dijkstra
b. Bellman-Ford
c. Depth-First Search
d. **Breadth-First Search**

a and b are less efficient and c cannot be used for finding shortest paths

Week 2 Quiz

In a weighted, directed graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the shortest path always remain the same?

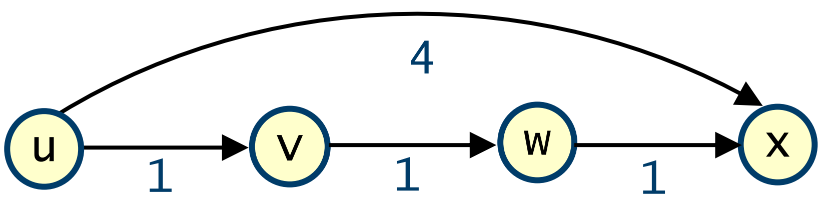
a. `true`
b. `false`

Week 2 Quiz

In a weighted, directed graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the shortest path always remain the same?

a. `true`
b. **`false`**

A path of length **k** will increase its distance by **k** (**1** is added to the weight of each edge in the path), so longer paths are penalised more



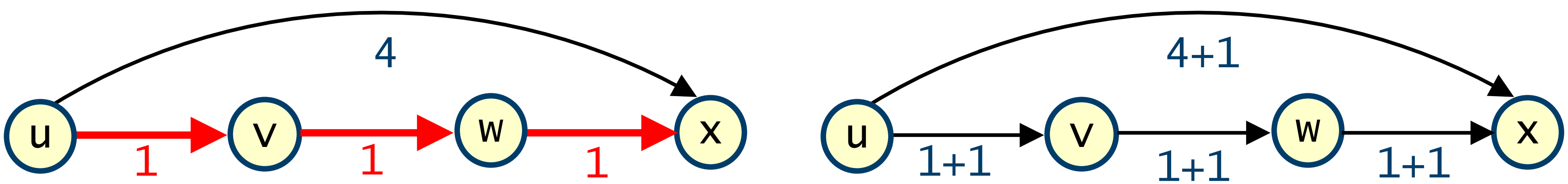
Week 2 Quiz

In a weighted, directed graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the shortest path always remain the same?

a. `true`

b. **`false`**

A path of length **k** will increase its distance by **k** (**1** is added to the weight of each edge in the path), so longer paths are penalised more



shortest path from u to x

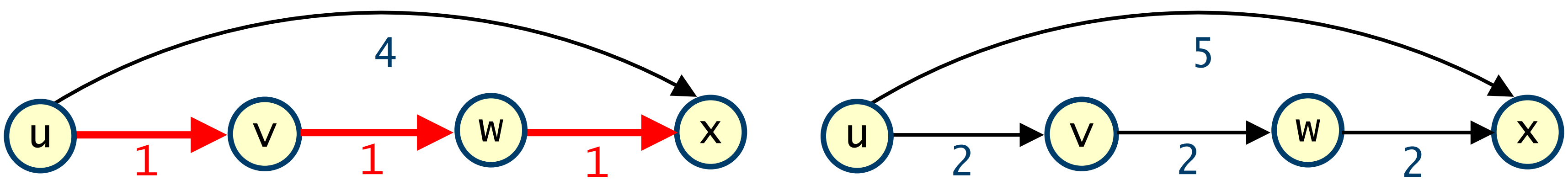
Week 2 Quiz

In a weighted, directed graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the shortest path always remain the same?

a. `true`

b. **`false`**

A path of length **k** will increase its distance by **k** (**1** is added to the weight of each edge in the path), so longer paths are penalised more



shortest path from u to x

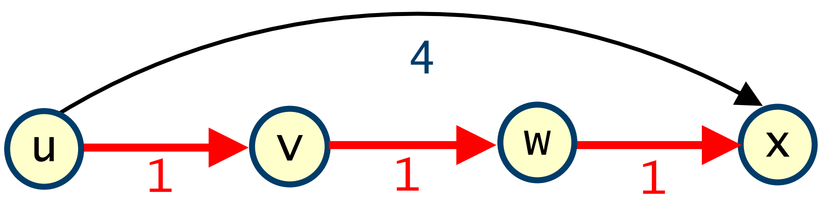
Week 2 Quiz

In a weighted, directed graph if we apply Dijkstra's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the shortest path always remain the same?

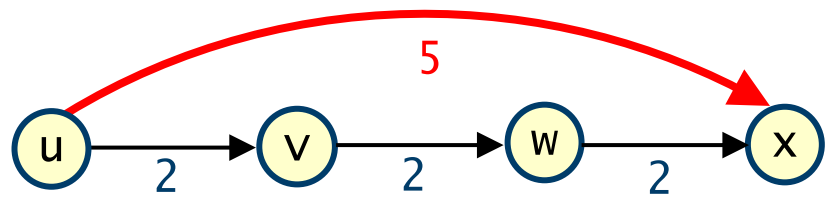
a. `true`

b. `false`

A path of length **k** will increase its distance by **k** (**1** is added to the weight of each edge in the path), so longer paths are penalised more



shortest path from u to x



shortest path from u to x

Week 3 Quiz

In a weighted, undirected graph if we apply Prim Jarnak's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the minimum weight spanning tree always remain the same?

a. `true`
b. `false`

Week 3 Quiz

In a weighted, undirected graph if we apply Prim Jarnak's algorithm to find the shortest path between two nodes. If we add **1** to all the edge weights, does the minimum weight spanning tree always remain the same?

a. **true**
b. false

Edges are chosen based on their weights and all increase by the same amount

- also all spanning trees have the same number of edges
    - if n vertices then $n-1$ edges in a spanning tree

So weight of each spanning tree increases by the same amount

Week 3 Quiz

Let G be an undirected connected graph with distinct edge weights, e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight.

Which of the following statements is false?

a. Every minimum spanning tree of G must contain the edge with weight e_{min}
b. If the edge with weight e_{max} is in a minimum spanning tree, then its removal must disconnect G
c. No minimum spanning tree contains the edge with weight e_{max}
d. G has a unique minimum spanning tree

Week 3 Quiz

Let G be an undirected connected graph with distinct edge weights, e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight.

Which of the following statements is `false`?

a. Every minimum spanning tree of G must contain the edge with weight e_{min}

`true`: this edge always added when one vertex of the edge is a `tv` and the other vetrex a `ntv`

Week 3 Quiz

Let G be an undirected connected graph with distinct edge weights, e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight.

Which of the following statements is `false`?

a. Every minimum spanning tree of G must contain the edge with weight e_{min}

b. If the edge with weight e_{max} is in a minimum spanning tree, then its removal must disconnect G

`true`: to include e_{max} must be no other option

- can make local optimal choices (as the Prim Jarnik algorithm does)

Week 3 Quiz

Let G be an undirected connected graph with distinct edge weights, e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight.

Which of the following statements is `false`?

a. Every minimum spanning tree of G must contain the edge with weight e_{min}
b. If the edge with weight e_{max} is in a minimum spanning tree, then its removal must disconnect G
c. **No minimum spanning tree contains the edge with weight e_{max}**

`false`: see case b which to a case when this edge is included

Week 3 Quiz

Let G be an undirected connected graph with distinct edge weights, e_{max} be the edge with maximum weight and e_{min} the edge with minimum weight.

Which of the following statements is `false`?

a. Every minimum spanning tree of G must contain the edge with weight e_{min}

b. If the edge with weight e_{max} is in a minimum spanning tree, then its removal must disconnect G

c. No minimum spanning tree contains the edge with weight e_{max}

d. G has a unique minimum spanning tree

`false`: can be more than one spanning tree

- as we have seen in the lectures

Week 3 Quiz

In a topological ordering, if `v` is a sink, it it true that `lab(u) < lab(v)` for all non-sink vertices `u`

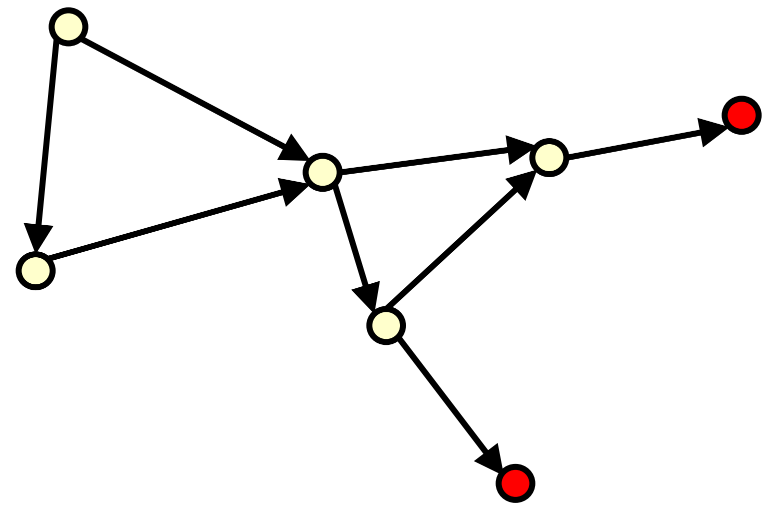
a. `true`

b. `false`

Week 3 Quiz

In a topological ordering, if **v** is a sink, it it true that
`lab(u) < lab(v)` for all non-sink vertices **u**

a. `true`
b. `false`



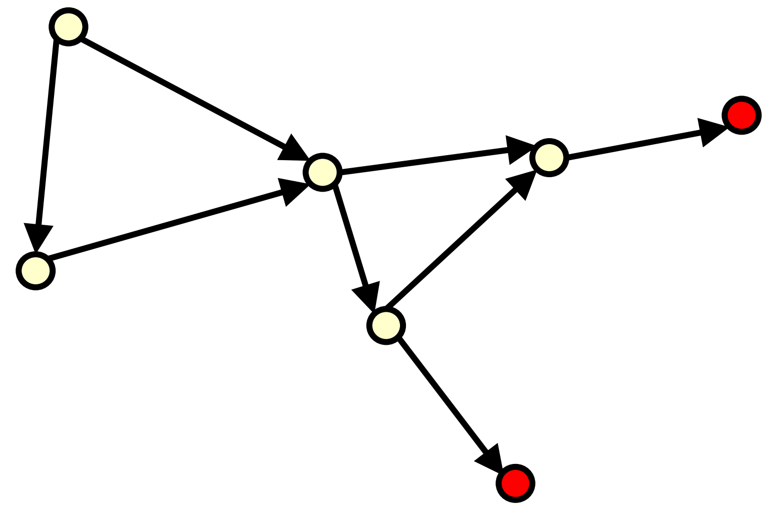
Week 3 Quiz

In a topological ordering, if **v** is a sink, it it true that

`lab(u) < lab(v)` for all non-sink vertices **u**

a. `true`

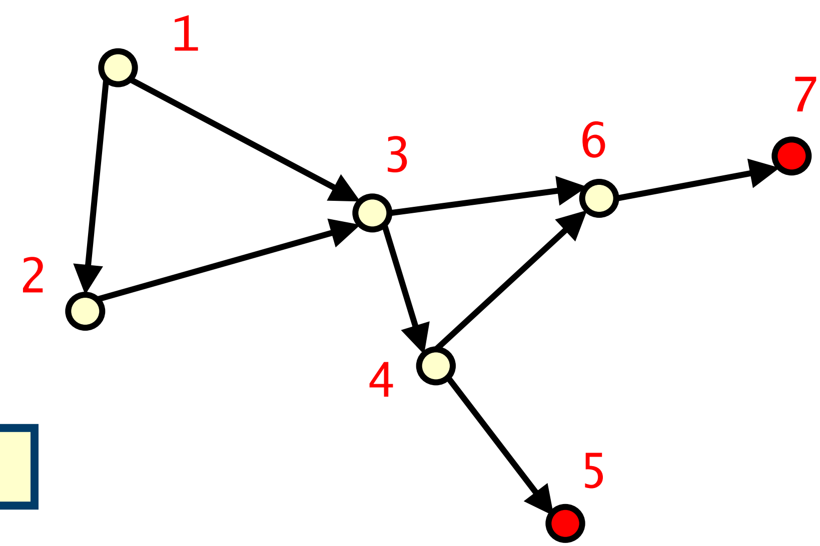
b. `false`



Week 3 Quiz

In a topological ordering, if **v** is a sink, it it true that
`lab(u) < lab(v)` for all non-sink vertices **u**

a. true
b. false



would be true if there was only one sink

Week 4 Quiz

Assuming the character frequencies have already been found, what is the complexity of building a Huffman tree if there are `m` characters?

a. O(m)
b. O(m log m)
c. O(m^2)
d. O(n^3)

Week 4 Quiz

Assuming the character frequencies have already been found, what is the complexity of building a Huffman tree if there are m characters?

a. O(m)
b. O(m log m)
c. $O(m^2)$
d. $O(n^3)$

Add leaf nodes with weights equal to character frequencies O(m)
Find two parentless nodes with smallest weights and add parent with weight equal to the sum of weights

- m times we remove and add elements to a heap
- with each operation O(log m)

Week 4 Quiz

What property of the dictionary holds throughout the running of the LZW algorithm?

a. If string s is in the dictionary, then all prefixes of s are in the dictionary
b. If string s is in the dictionary, then no prefixes of s are in the dictionary
c. All strings of length k are in the dictionary when the codeword length is k
d. No code word is a prefix of another code word

Week 4 Quiz

What property of the dictionary holds throughout the running of the LZW algorithm?

a. **If string s is in the dictionary, then all prefixes of s are in the dictionary**
b. If string s is in the dictionary, then no prefixes of s are in the dictionary
c. All strings of length k are in the dictionary when the codeword length is k
d. No code word is a prefix of another code word

```
identify the longest string s, starting at position i of text t
that is represented in the dictionary d;
...
add string s+c to dictionary d, paired with next available codeword;
```

Week 4 Quiz

In the recurrence relation for finding the optimal string distance when transforming the string `x` into the string `y`. When the characters do not match the distance `d(i,j)` equals:

`1 + min{ d(i,j-1), d(i-1,j), d(i-1,j-1) }`

What does the case `d(i-1,j-1)` correspond to?

a. Deleting an element from x
b. Inserting an element in x
c. **Substituting an element in x**
d. Inserting an element in y

After substitution matched element of x with that of y

Week 5 Quiz

In the table is required by the KMP algorithm what is included in entry `i` when searching for the pattern/string `s`?

a. The shortest border of the substring `s[0..i-1]`
b. The longest border of the substring `s[0..i-1]`
c. The last occurrence of the character `s[i]`
d. The first occurrence of the character `s[i]`

Week 5 Quiz

In the table is required by the KMP algorithm what is included in entry `i` when searching for the pattern/string `s`?

a. The shortest border of the substring `s[0..i-1]`
b. **The longest border of the substring `s[0..i-1]`**
c. The last occurrence of the character `s[i]`
d. The first occurrence of the character `s[i]`

Look at how the algorithm works

Week 5 Quiz

What is the length of the longest border of **abcabca**?

a. 2

b. 4

c. 5

Week 5 Quiz

What is the length of the longest border of abcabca?

a. 2 not a border: abcabca abcabca

b. 4 longest border: abcabca abcabca

c. 5 not a border: abcabca abcabca

Week 5 Quiz

In the table is required by the Boyer Moore algorithm what is included in entry `i` when searching for the pattern/string `s` in the text `t`?

a. The shortest border of the substring `t[0..i-1]`
b. The longest border of the substring `t[0..i-1]`
c. The last occurrence of the character `t[i]`
d. The first occurrence of the character `t[i]`

Week 5 Quiz

In the table is required by the Boyer Moore algorithm what is included in entry `i` when searching for the pattern/string `s`?

a. The shortest border of the substring `t[0..i-1]`
b. The longest border of the substring `t[0..i-1]`
c. The last occurrence of the character `t[i]`
d. The first occurrence of the character `t[i]`

Look back at the slides on how the algorithm works

Week 6 Quiz

Suppose problem X is in P, problem Y is in class NP, and there is a polynomial reduction from X to Y. Which of the following is true?

a. Problem Y is in class P
b. Problem Y is NP-complete
c. Both of the above
d. Neither of the above

Week 6 Quiz

Suppose problem X is in P, problem Y is in class NP, and there is a polynomial reduction from X to Y. Which of the following is true?

a. Problem Y is in class P
b. Problem Y is NP-complete
c. Both of the above
d. Neither of the above

All this shows is that Y is “at least as hard” as X, but might map problems of X to easy problems of Y

Week 6 Quiz

Suppose problem X is NP-complete, problem Y is in class NP, and there is a polynomial reduction from X to Y. Which of the following is true?

a. Problem Y is in class P
b. Problem Y is NP-complete
c. Both of the above
d. None of the above

Week 6 Quiz

Suppose problem X is NP-complete, problem Y is in class NP, and there is a polynomial reduction from X to Y. Which of the following is true?

a. Problem Y is in class P
b. Problem Y is NP-complete
c. Both of the above
d. None of the above

- if X in NP-complete, then we can reduce any NP problem to X in polynomial time
- by the hypothesis we can reduce X to Y in polynomial time
- therefore we can reduce any NP problem to Y in polynomial time
  - first reduce it to X and then from X reduce to Y
- and since Y is in NP, it follows Y is NP-complete

Week 6 Quiz

Suppose problem X is `NP-complete`, problem Y is in class P, and there is a polynomial reduction from X to Y. If Z is `NP-complete`, which of the following is true?

a. Problem Z is in class P
b. Problem Z is in the class NP
c. Both of the above
d. None of the above

Week 6 Quiz

Suppose problem X is `NP-complete`, problem Y is in class P, and there is a polynomial reduction from X to Y. If Z is `NP-complete`, which of the following is true?

a. Problem Z is in class P
b. Problem Z is in the class NP
c. Both of the above
d. None of the above

– we can solve X in polynomial time by reducing to Y and solving Y
– since X is NP-complete this means all NP-complete problems can be solved in polynomial time including Z
– since Z is NP-complete, by definition it is in NP

Week 7 Quiz

Which of the following strings is a member of the language over **{a,b}** defined by the regular expression **(aa|ba)*(bb)***?

a. aabbba
b. aaaabb
c. babbaa
d. bbaa

Week 7 Quiz

Which of the following strings is a member of the language over **{a,b}** defined by the regular expression **(aa|ba)*(bb)***?

a. aabbba
- ends ba and these is only achievable when aa or ba precedes the ba

b. aaaabb
- aaaa comes from aa twice followed by bb which comes from bb once

c. babbaa
- ends aa and these is only achievable when aa or ba precedes the aa

d. bbaa
- ends aa and these is only achievable when aa or ba precedes the aa

Week 7 Quiz

Over the alphabet **{a,b}**, which of the following the regular expressions represents the language which consists of strings that start and end with different symbols.

a. a(a|b)*b
b. b(a|b)*a
c. (a(a|b)*b) | (b(a|b)*a)
d. All of the above

Week 7 Quiz

Over the alphabet **{a,b}**, which of the following the regular expressions represents the language which consists of all strings that start and end with different symbols.

a. a(a|b)*b
b. b(a|b)*a
c. **(a(a|b)*b) | (b(a|b)*a)**
d. All of the above

a (b) are strings that start with **a (b)** and end with **b (a)**

Week 7 Quiz

Over the alphabet **{a,b}**, which of the following the regular expressions represents the language which consists of strings that starts with **ab** and ends with **ba**.

a. `aba*b*ba`
b. `ab(ab)*ba`
c. `ab (a|b)*ba`
d. All of the above

Week 7 Quiz

Over the alphabet **{a,b}**, which of the following the regular expressions represents the language which consists of strings that starts with **ab** and ends with **ba**.

a. aba*b*ba starts with ab, then ≥0 as then ≥0 bs, ends with ba
b. ab(ab)*ba starts with ab, then ≥0 abs, ends with ba
c. ab (a|b)*ba all strings that starts with ab and end with ba
d. All of the above

question should have said consists of **all** strings that ....

will mark both c and d as correct (ignore moodle I will export the individual marks for quiz questions and fix)

Week 8 Quiz

Which of the following can be computed by a deterministic finite state automaton? (Select all that are computable).

a. Strings which have more as than bs
b. Strings which have no single as (i.e. any a is either preceded or followed by another a)
c. Strings that start and end with the same character
d. Strings that start with a certain number of as end with the same number of as

Week 8 Quiz

Which of the following can be computed by a deterministic finite state automaton? (Select all that are computable).

a. Strings which have more as than bs
b. **Strings which have no single as (i.e. any a is either preceded or followed by another a)**
c. **Strings that start and end with the same character**
d. Strings that start with a certain number of as end with the same number of as

- b and c can be expressed by the regular expressions `(b*aaa*b*)*` and `(a(a|b)*a)|(b(a|b)*b)`
- a and d both require counting the number of a certain character and keeping track of this, but we only have finitely many states

Week 8 Quiz

Does nondeterminism increase the languages that can be expressed by finite state automata?

a. yes
b. no

Week 8 Quiz

Does nondeterminism increase the languages that can be expressed by finite state automata?

a. yes
b. no

For any NFA one can build (using the subset construction) a DFA that recognises the same language

- this can cause an exponential blow up in the state space

Week 8 Quiz

Does nondeterminism increase the languages that can be expressed by pushdown automata?

a. yes
b. no

Week 8 Quiz

Does nondeterminism increase the languages that can be expressed by pushdown automata?

a. **yes**

b. no

For example for the language of palindromes, we need to guess when we have reached the "middle" of the input