

Revue des SDD

30 septembre 2024 12:55

TDA: liste

• Std lib: `std::list`

	Complexité (liste chaînée)	Complexité (tab)
• <code>push_back(val)</code>	$O(1)$	$O(1)$ AMORTI
<code>push_front(val)</code>	$O(1)$	$O(1)$ AMORTI
<code>pop_back()</code>	$O(1)$	$O(1)$
<code>pop_front()</code>	$O(1)$	$O(1)$

Autres

• <code>size()</code>	$O(1)$ si var membre, $O(n)$ sinon	$O(1)$
<code>at(i), [i]</code>	$O(n)$	$O(1)$
<code>insert(it, val)</code>	$O(1)$	$O(n)$
<code>erase(it)</code>	$O(1)$	$O(n)$

TDA: file

Std lib: `std::queue`

• FIFO 

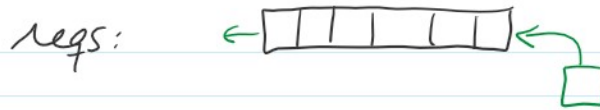
• `push_back(val)`

`pop_front()`

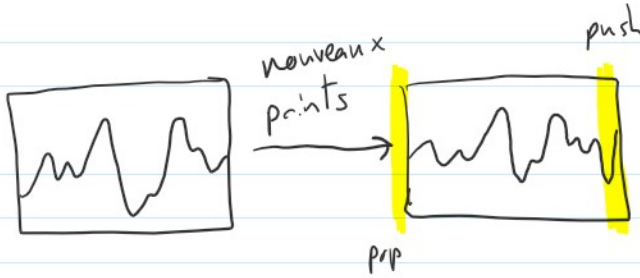
`front()`

`size()`

ex: • gestion de requêtes en ordre d'arrivée




- buffers (tampon de données)



Défi: get_min, get_max en temps $O(1)$ à tout moment

TDA: pile

Std lib: `std::stack`

- Last-in first-out (Lifo) 

- `push(val)` : empiler `push_back(val)`
`pop()` : dépiler `pop_back()`
`top()` : voir sommet `back()`

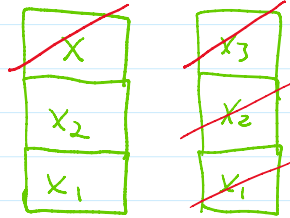
ex: • gestion de la mémoire auto

```
int incl(int x) {  
    return x+1;  
}  
int x1 = 0;  
int x2 = 10;
```

```

int x1 = 0;
int x2 = 10;
while (x1 < x2) {
    x1 = inc(x1);
    int x3 = x1;
    cout << x3;
}

```

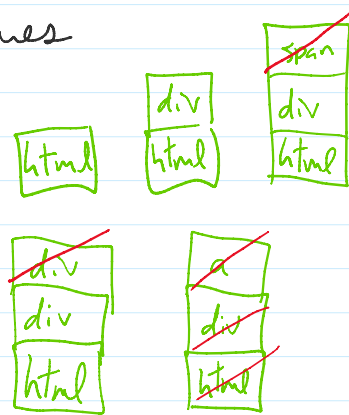


• valider des documents hiérarchiques

```

<html>
  <div>
    <span>allo </span>
    <div> salut </div>
  </div>
  <a href='...'> ... </a>
</html>

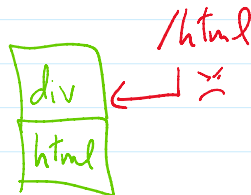
```



```

<html>
<div>
</html>
</div>

```



• expression bien parenthésée

• Voir exercices!