IOUQueue (Deque)

IOUQueue[*1] は、Data.Vector.Unboxed.Mutable を用いた Deque である。大規模な BFS などを高速に実行することを意図して作成した。

ひとつ、または、複数の値を追加する関数に加え、BFS では探索における距離などをタプルにして要素に付加することも多いことから、`[a]` 型のリストと定数 `b` を引数にとって、`(a, b)` 型の要素を push する関数も用意した (pushBackListWithD / pushFrontListWithD)。

コンストラクタ

長さを指定して Deqeue を生成する。長さは足りなくなった場合に自動的に拡張される。

```
newUDeque :: (Unbox a) => Int -> IO (IOUQueue a)
```

制約

- $0 \ \leq n \leq 10^8$

計算量

- O(n)

null

Deque が空かどうかを判定する。戻り値は `IO Bool` ではなく `Bool` である点に注意。

```
nullUQ :: (Unbox a) => IOUQueue a -> Bool
```

計算量

- O(1)

length

Deque の長さを返す。戻り値は `IO Int` ではなく `Int`

[*1] https://github.com/unnohideyuki/AtHaskell/blob/master/snippets/IOUQueue.hs

```
lengthUQ :: (Unbox a) => IOUQueue a -> Int
```

計算量

- O(1)

grow

Deque の長さを倍にする。push の際に必要に応じて用いられるので、ユーザが直接用いる必要はない。

```
growUQ :: (Unbox a) => IOUQueue a -> IO (IOUQueue a)
```

計算量

- O(n)

popFront / popBack

Deque の先頭または末尾から要素をひとつ取り出す。

```
popFrontUQ :: (Unbox a) => IOUQueue a -> IO (IOUQueue a, a)
popBackUQ :: (Unbox a) => IOUQueue a -> IO (IOUQueue a, a)
```

計算量

- O(1)

pushFront /pushBack

Deque の先頭または末尾に複数の要素をひとつ追加する。

```
pushFrontUQ :: (Unbox a) => IOUQueue a -> a -> IO (IOUQueue a)
pushBackUQ :: (Unbox a) => IOUQueue a -> a -> IO (IOUQueue a)
```

計算量

- O(1)

pushFrontList / pushBackList

Deque の先頭または末尾に複数の要素を複数追加する。

```
pushFrontListUQ :: (Unbox a) => IOUQueue a -> [a] -> IO (IOUQueue a)
pushBackListUQ :: (Unbox a) => IOUQueue a -> [a] -> IO (IOUQueue a)
```

計算量

- O(n)

pushFrontListWithD / pushBackListWithD

Deque の先頭または末尾に、定数を付加した要素を複数追加する。`[a]` と `b` をとり、`(a,b)` 型の要素を複数追加する。

```
pushFrontListWithDUQ :: (Unbox a, Unbox b) => IOUQueue (a, b) -> [a] -> b -> IO (IOUQueue (a, b))
pushBackListWithDUQ :: (Unbox a, Unbox b) => IOUQueue (a, b) -> [a] -> b -> IO (IOUQueue (a, b))
```

計算量

- O(n)