BAG type defenition

values:

BAG

Represantation:

stuff : Elements* ← List of elements

Elements : Tuple (int, int)

elem, frequency

Operations:

1. searchBag : A = (bag:BAG, e : ℤ, [alt, searched : ℕ)

(linearsearch) pre = (e = e', bag = bag')

post = (pre ∧

searched := -1 ∧ (ind)searched = $\underset{i=1}{\overset{stuff.count}{search}}$ (stuff[i].item1 == e)

2. insertElement:

$A = (bag: BAG, e: \mathbb{Z})$

$pre = (e = e', bag = bag')$

$post = (searched = searchBag(e) \wedge$

$searched = -1 \Rightarrow stuff \overset{(add)}{\oplus} (e, 1)$

$\vee\ stuff[searched] = (e, stuff[searched].item2 + 1))$

3. removeElement:

$A = (bag: BAG, e: \mathbb{Z})$

$pre = (e = e', bag = bag', bag.stuff.count > 0)$

$post = (searched = searchBag(e) \wedge$

$searched \neq -1 \wedge$

$stuff[searched].item2 == 1 \Rightarrow stuff \overset{(removeAt)}{\ominus} stuff[searched]$

$\vee\ stuff[searched] = (e, stuff[searched].item2 - 1))$

4. frequency

$A = (bag : BAG, e : \mathbb{Z}, f : \mathbb{N})$

$pre = (bag = bag', e = e')$

$post = (pre \wedge searched = searchBag(e) \wedge$

$searched = -1 \Rightarrow f = 0$

$\vee\ f = stuff[searched].item2$

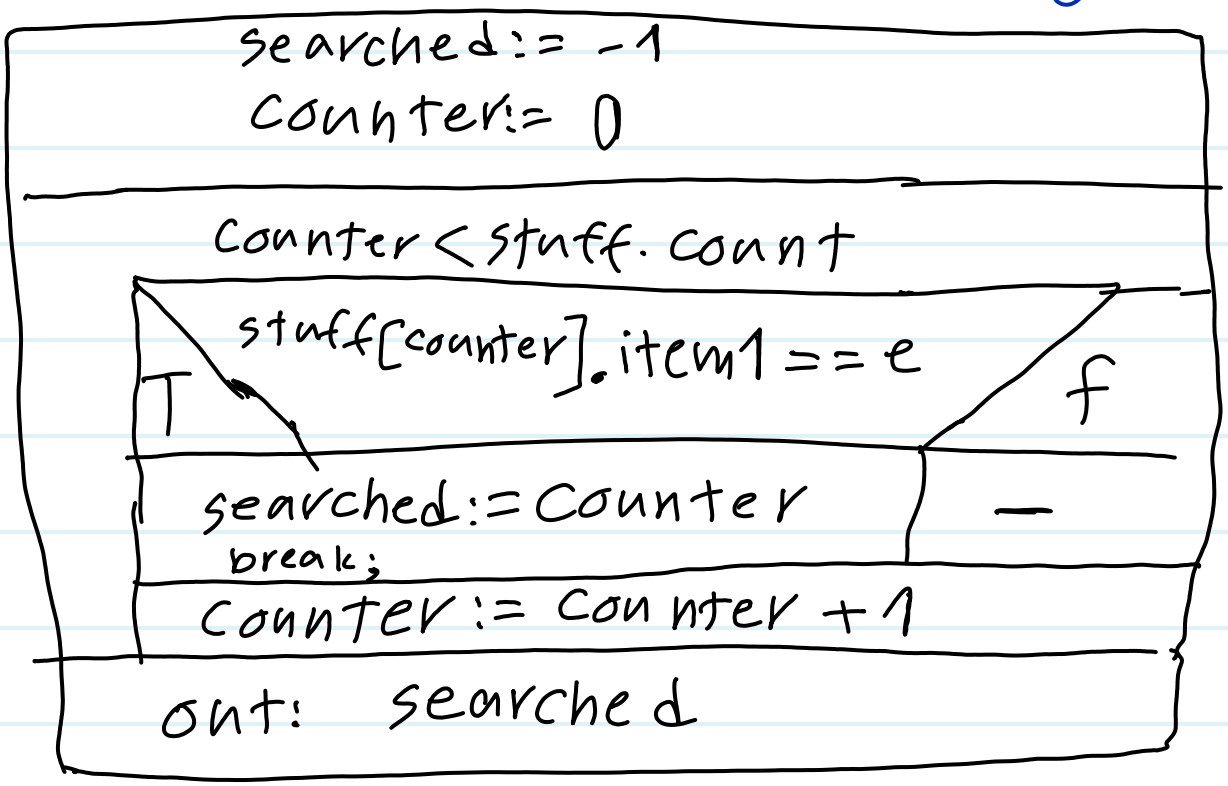
5. largestElement :

$A = (bag : BAG, largest : \mathbb{Z})$

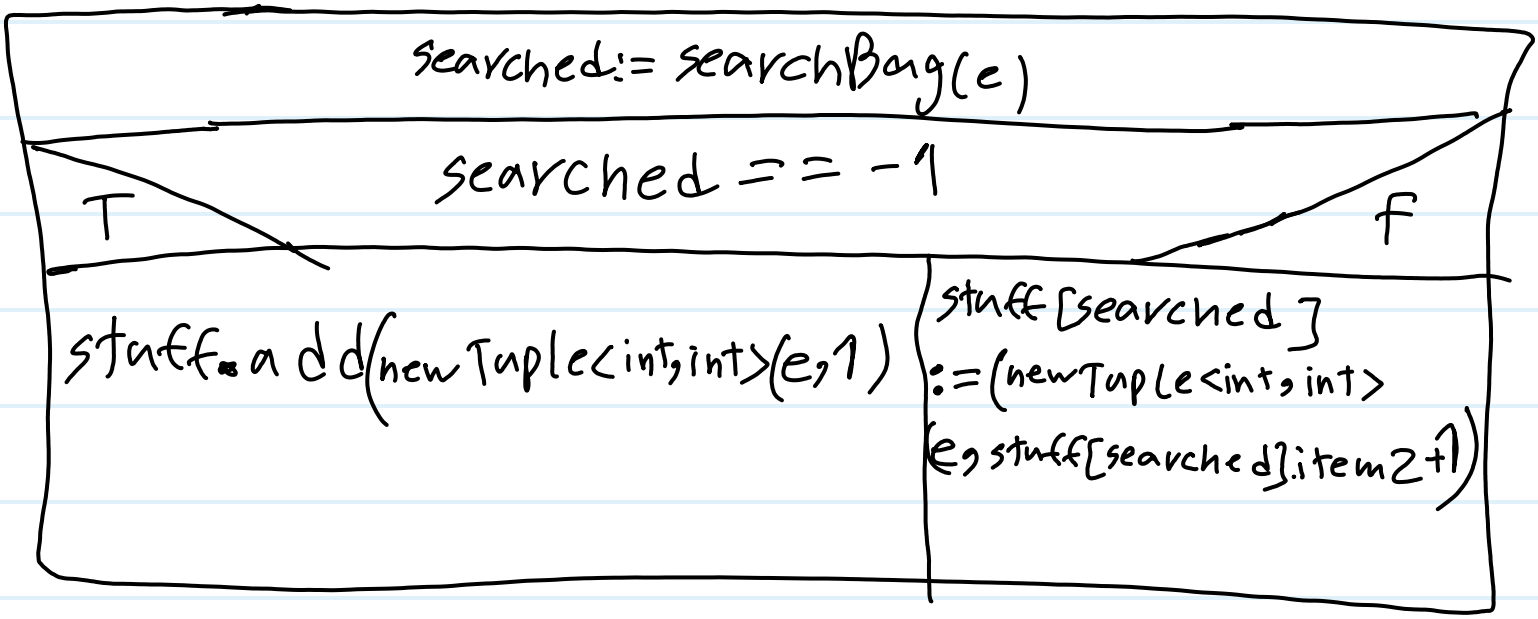
$pre = (bag = bag', bag.stuff.count > 0)$

$post = (pre \wedge (max, largest) = (\max_{i=1}^{bag.stuff.count} stuff[i].item2, stuff[i].item1))$

implementation: 1. searchBag(e:int):int

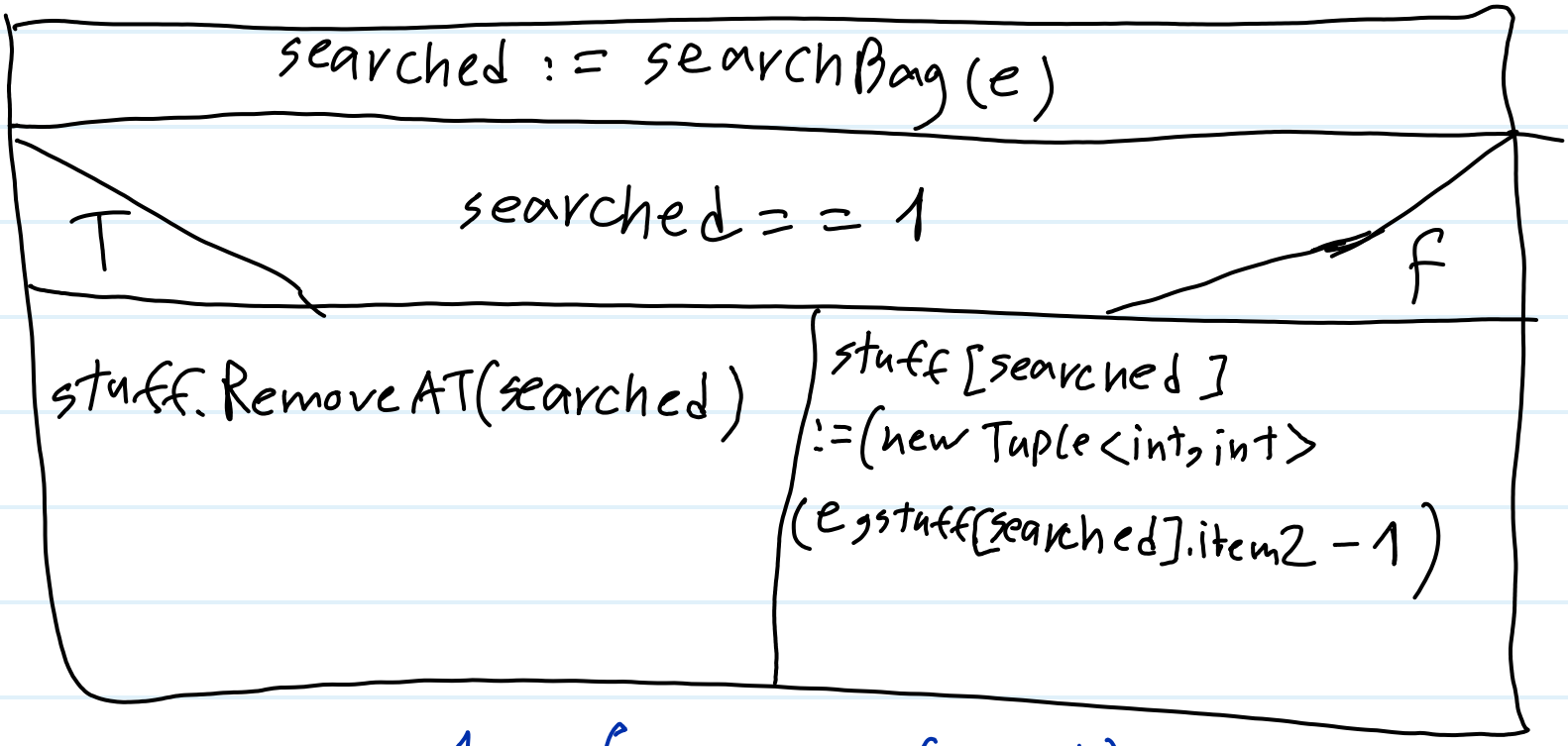


2. insertBag(e:int):void

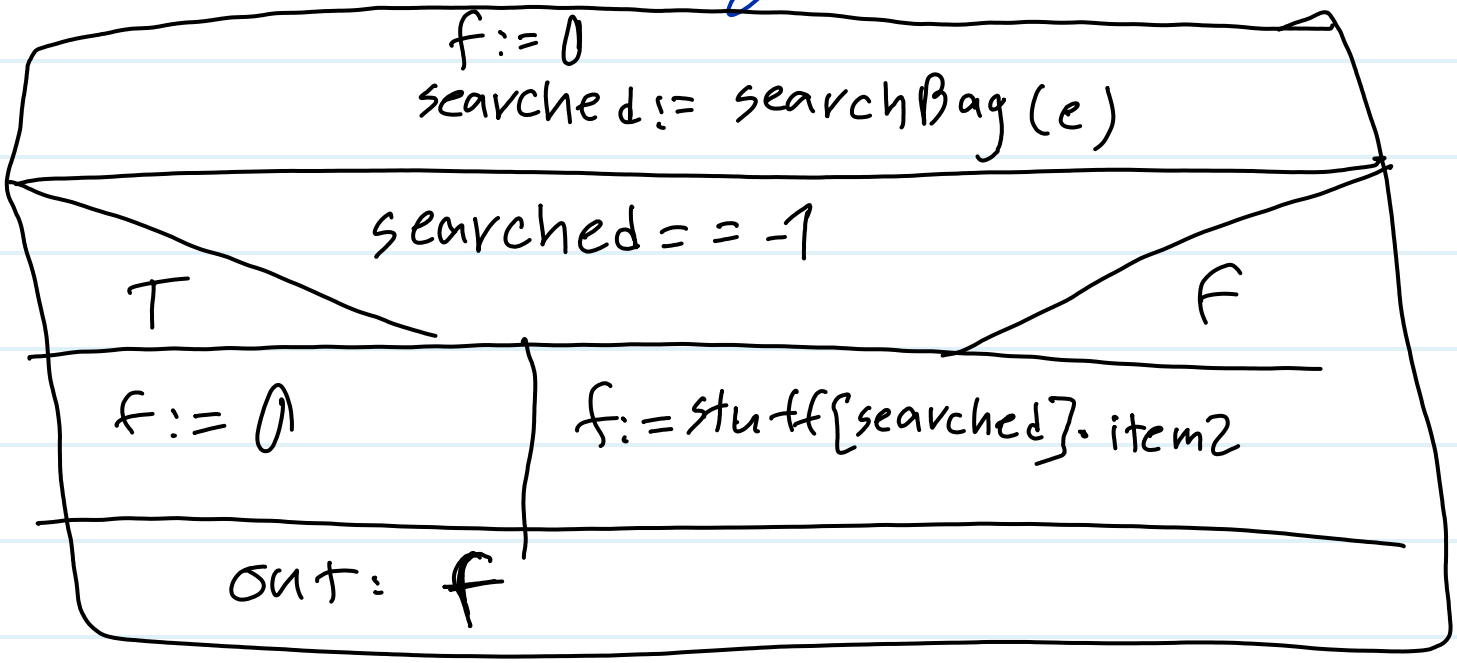


Tuples cant change their item1, item2 values so they need to be overwritten

3. removeBag(e:int):void



4. frequency(e:int):int



5. largestElement(): int