# a.py

---

```
def compute_normalization(self, e_sentence, f_sentence):
```

The above function in `a.py` should be defined according to the following formula,

$$\forall_{e \in \boldsymbol{e}} \quad z_e = \sum_{f \in \boldsymbol{f}} t(e|f) \tag{1}$$

where $t(\ldots)$ is the conditional distribution `self.t`, $\boldsymbol{e}$ is the English sentence `e_sentence`, $\boldsymbol{f}$ is the corresponding foreign sentence `f_sentence`, $z_e$ is the computed normalization constant for a particular English word $e$ in $\boldsymbol{e}$, and $\boldsymbol{z}$ is a dictionary that maps from each English word $e$ to the corresponding calculated normalization constant $z_e$ for all $e$ in $\boldsymbol{e}$.

The function should return $\boldsymbol{z}$, the dictionary of computed values.

---

```
def update_counts(self, e_sentence, f_sentence, counts, z)
```

The above function in `a.py` should be defined according to the following formula,

$$\forall_{e \in \boldsymbol{e}} \forall_{f \in \boldsymbol{f}} \quad c(e|f) = c(e|f) + \frac{t(e|f)}{z_e} \tag{2}$$

where $c(\ldots)$ is the conditional distribution `counts`, $\boldsymbol{e}$ is the English sentence `e_sentence`, $\boldsymbol{f}$ is the corresponding foreign sentence `f_sentence`, $z_e$ is the normalization constant for a particular English word $e$ in $\boldsymbol{e}$, and $\boldsymbol{z}$ is the dictionary `z` that maps from each English word $e$ to the corresponding calculated normalization constant $z_e$ for all $e$ in $\boldsymbol{e}$.

The function will not return a value. Instead, it will have the side effect of causing values in the conditional distribution `counts` to be updated.

---

```
def update_totals(self, e_sentence, f_sentence, totals, z)
```

The above function in `a.py` should be defined according to the following formula,

$$\forall_{e \in \boldsymbol{e}} \forall_{f \in \boldsymbol{f}} \quad \texttt{totals}[f] = \texttt{totals}[f] + \frac{t(e|f)}{z_e} \tag{3}$$

where `totals` is a dictionary that maps from foreign words to floating point values, and the other items are as defined above.

The function will not return a value. Instead, it will have the side effect of causing values in the dictionary `totals` to be updated.

---

```
def update_probabilities(self, counts, totals)
```

The above function in `a.py` should be defined according to the following formula,

$$\forall_{e \in \boldsymbol{e}} \forall_{f \in \boldsymbol{f}} \quad t(e|f) = \frac{c(e|f)}{\texttt{totals}[f]} \tag{4}$$

where $c(\ldots)$ is the conditional distribution `counts`, $t(\ldots)$ is the conditional distribution `self.t`, and the other items are as defined above.

The function will not return a value. Instead, it will have the side effect of causing values in the dictionary `self.t` to be updated.

```
def initialize_totals(self)
```

The above function in `a.py` should be defined according to the following formula,

$$\forall_{f \in \boldsymbol{F}} \ \texttt{totals}[f] = 0.0 \tag{5}$$

where $\boldsymbol{F}$ is the foreign vocabulary.

The function should return the dictionary `totals`.