

## Prompt Template

### Role to be played :

You are a leading expert on this topic. Given the existing function, please generate an optimized version.

### VM Task Description:

1. Calculate and return the priority score for adding a specific item to a bin based on available CPU and MEM resources.
2. Args: bin (tuple): representing the bin's available resources. item (tuple): representing the item's resource requirements.
3. Returns:  
int: The total score for placing the item in the current bin.  
A higher score indicates a better fit based on current available resources.

### Code Snippet: (policy $\pi$ in code form)

```
import numpy as np
def priority_v0(bin, item):
    score = -(bin[0] - item[0])
    return score
def priority_v1(bin, item):
    """Improved version of priority_v0."""
```



## Improvement via LLMs Generate policy $\pi'$ in code form

### Code Snippet: (policy $\pi'$ )

```
import numpy as np
def priority (bin, item):
    .....
    size_factor = (cpu_needed *
mem_needed) / (cpu_resource * mem_resource)
    min_weighted_diff *= (1 - size_factor)
    score = min_weighted_diff
    return score
```

## Code Database

### candidate code 1

```
dynamic_factor = (bin[0] / (bin[0] + bin[1]...
```

### candidate code 2

```
import numpy as np;
def priority (bin, item):...
```

...

### candidate code N

```
adjusted_cpu_weight = \
cpu_weight * dynam...
```

Evaluate and insert the generated code

### Code Snippet:(optimal policy $\pi^*$ )

```
# Calculate remaining space as a product of
differences, normalized by the bin's total
resources
```

```
remaining_space = (cpu_diff * mem_diff)
/ (cpu_resource * mem_resource)
```

```
# Adjust min_weighted_diff based on
remaining space
```

```
min_weighted_diff *= (1 -
remaining_space)
```

```
# Calculate the size ratio of the item to the
bin to assess item-to-bin size proportion
```

```
item_size = cpu_needed * mem_needed
bin_size = cpu_resource *
mem_resource; size_ratio = item_size /
bin_size
```

termination  
condition

Y Output the Best Policy

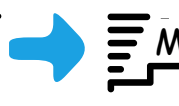
N Continue



evaluating



sorting



top-M  
selection

Plug Top-M Policies into the Prompt