CS630 Graduate Algorithms

October 3, 2024

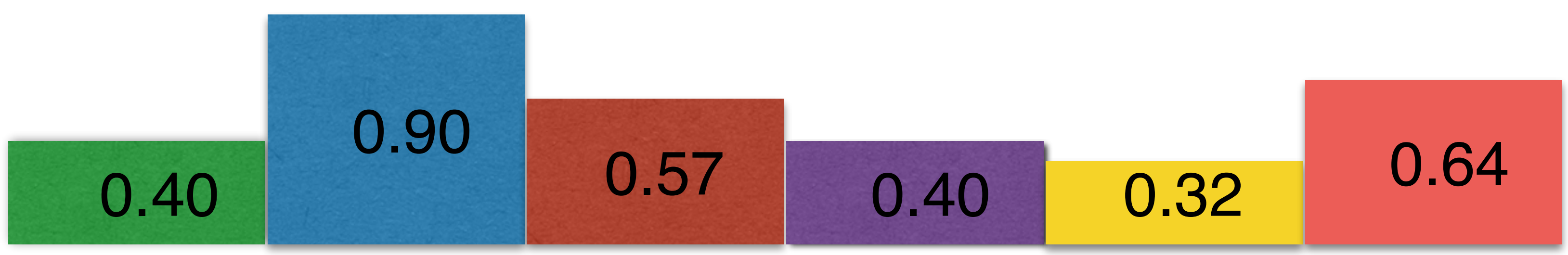
by Dora Erdos and Jeffrey Considine

- Bin packing and related problems

Bin Packing

Suppose you have a multiset of rational numbers $S = \{s_1, s_2, \ldots, s_n\}$ where $0 < s_i < = 1$ and you want to pack them into a minimal number of bins of size one. That is, you want to partition S such that the sum of each partition is at most one.

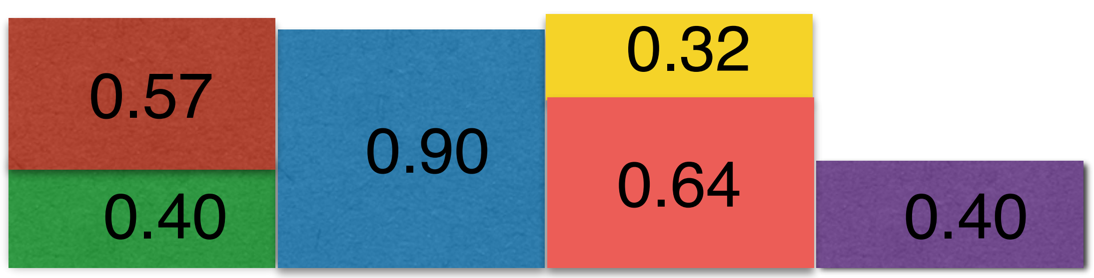
- What's the minimum bins you can pack all of S into?
- What partitioning of S will let you do that?



Bin Packing

Suppose you have a multiset of rational numbers $S = \{s_1, s_2, \ldots, s_n\}$ and you want to pack them into a minimal number of bins of size one. That is, you want to partition S such that the sum of each partition is at most one.

- What's the minimum bins you can pack all of S into?
- What partitioning of S will let you do that?



- Is this a hard problem in general?

Bin Packing is NP-Complete

$SAT \leq_P SUBSETSUM \leq_P PARTITION \leq_P BINPACKING$

SAT:

- Is this boolean formula (or circuit) satisfiable?
- Is there a set of inputs making it output true?
- OG NP-Complete problem

Subset-SUM:

- Given a multi-set of integers S, is there a subset of them adding up to T?
- One of Karp's list of 21 NP-Complete problems

Partition:

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2}\sum s_i$?

Bin packing:

- Given a multiset of rational numbers $S = \{s_1, s_2, \ldots, s_n\}$, can you pack them into k bins of size one?

Bin Packing is NP-Complete

$$SAT \leq_P SUBSETSUM \leq_P PARTITION \leq_P BINPACKING$$

All of these problems are NP-Complete.

- No known polynomial time algorithms.
- No expectations of a general fast algorithm any time soon.

Next Fit Bin Packing - Approximation Algorithms?

How can we bound the approximation ratio?

- Lower bound for space required?

Next Fit Bin Packing - Approximation Algorithms?

How can we bound the approximation ratio?

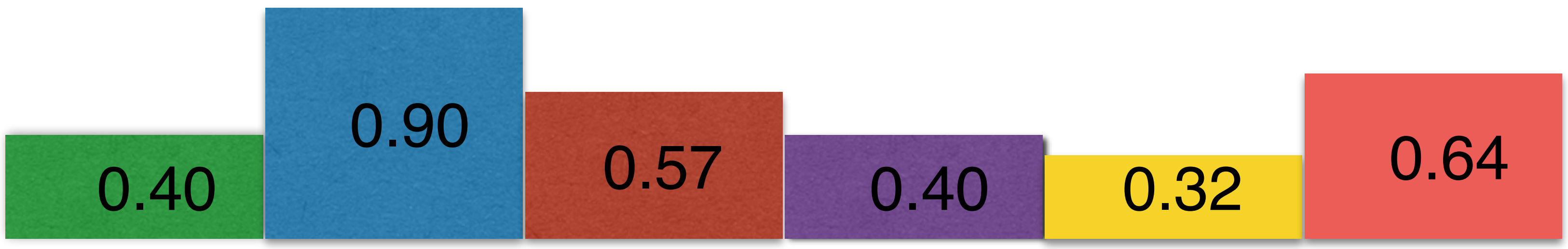
- Lower bound for space required?

- Easy lower bound is $\left\lceil \sum s_i \right\rceil$

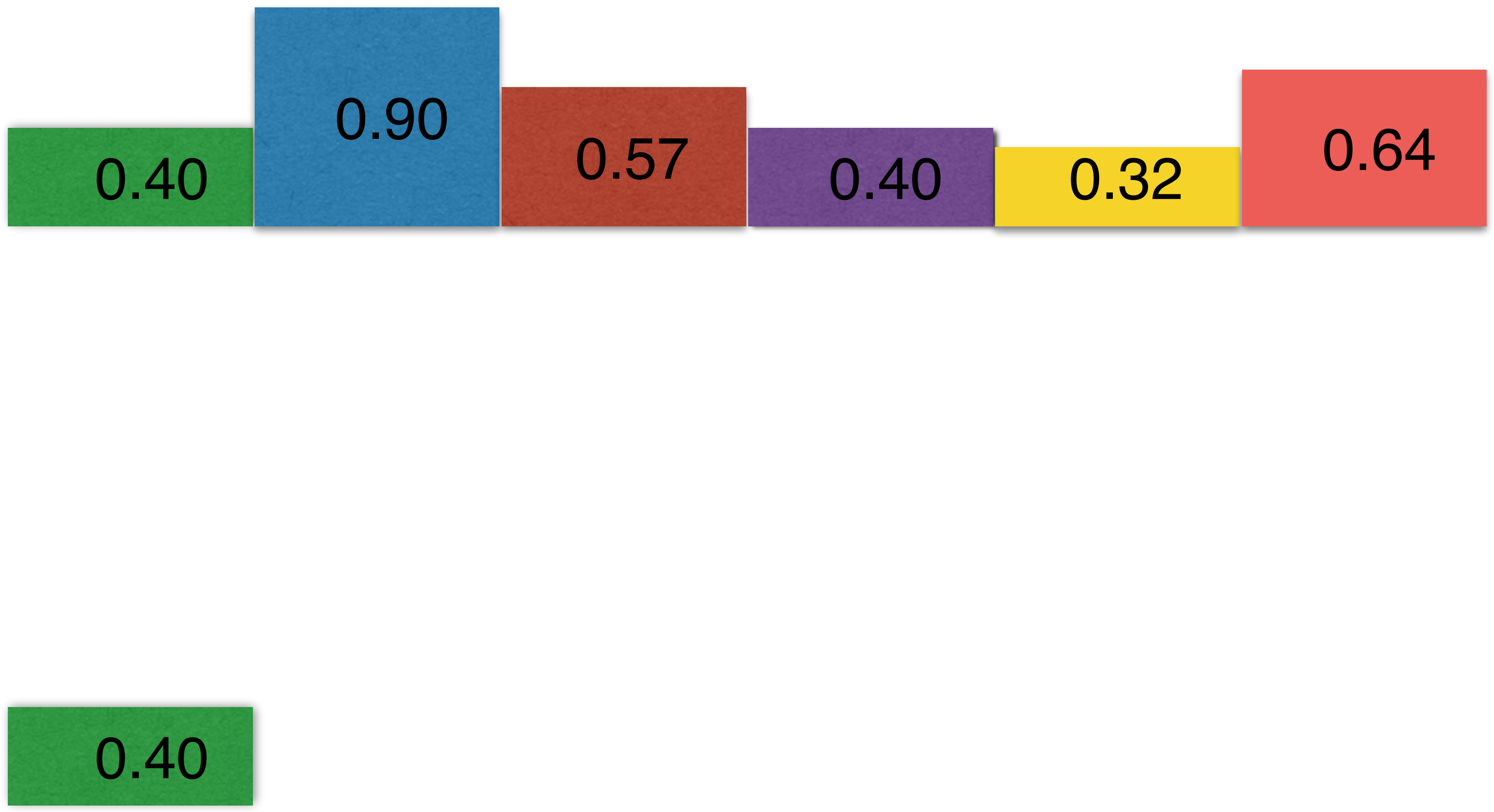
Next Fit Bin Packing

```
Start with one empty bin and call it “current”.
For each item:
    If there is room in the current bin,
        Put the item in the current bin.
    Otherwise,
        Start a new bin and make it current.
        Put the item in the new current bin.
```

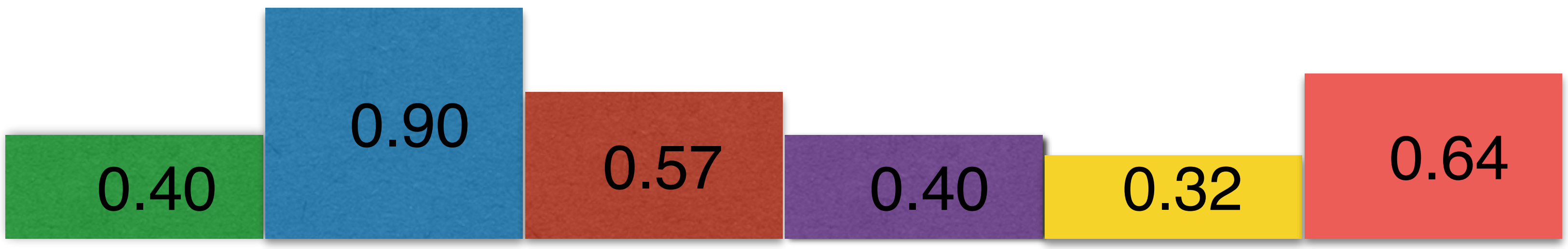
Next Fit Bin Packing

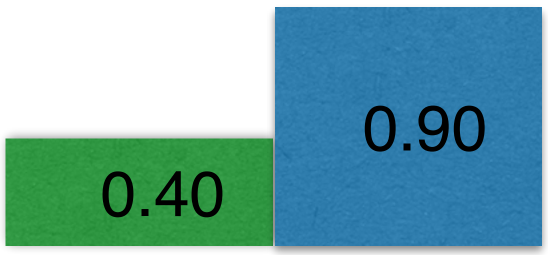


Next Fit Bin Packing

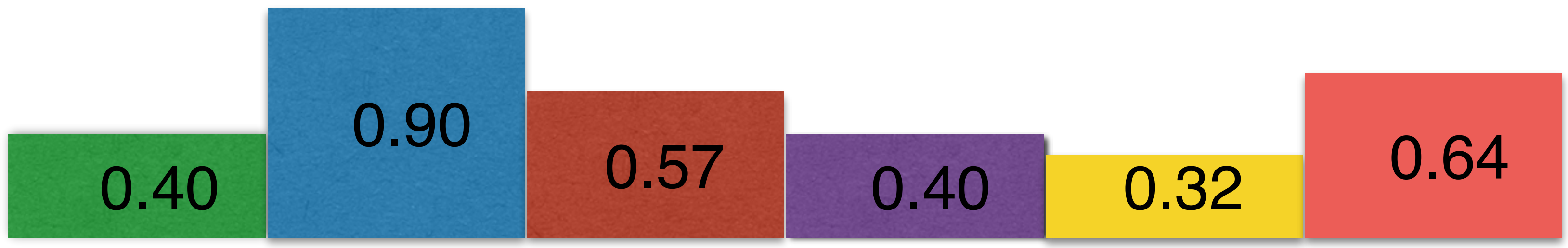


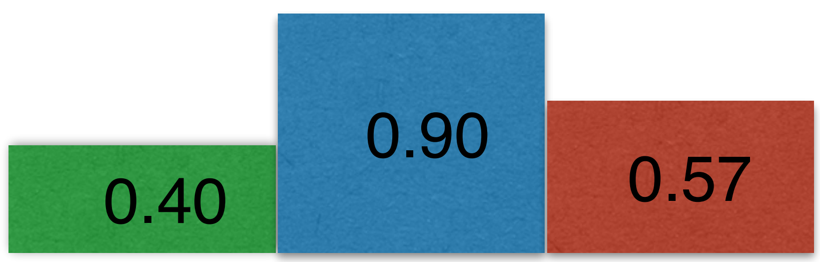
Next Fit Bin Packing



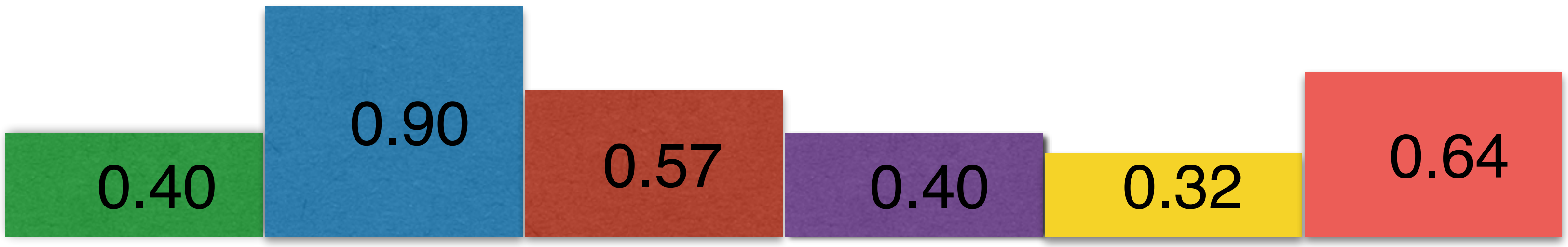


Next Fit Bin Packing

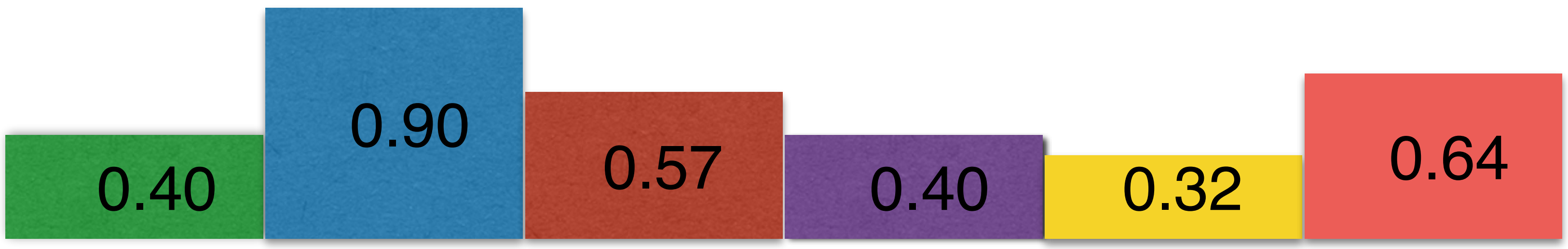


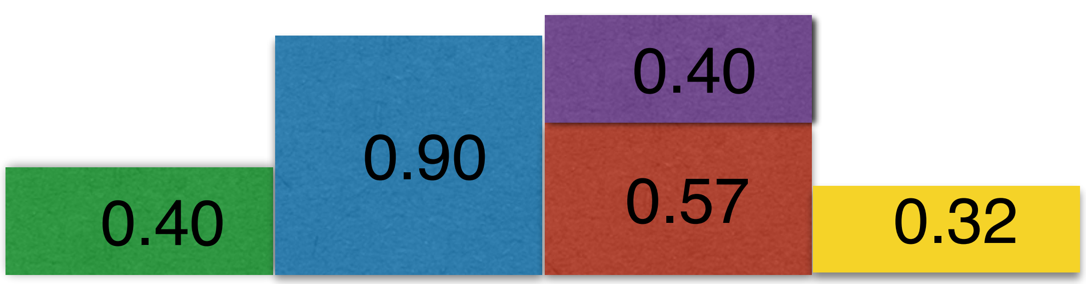


Next Fit Bin Packing

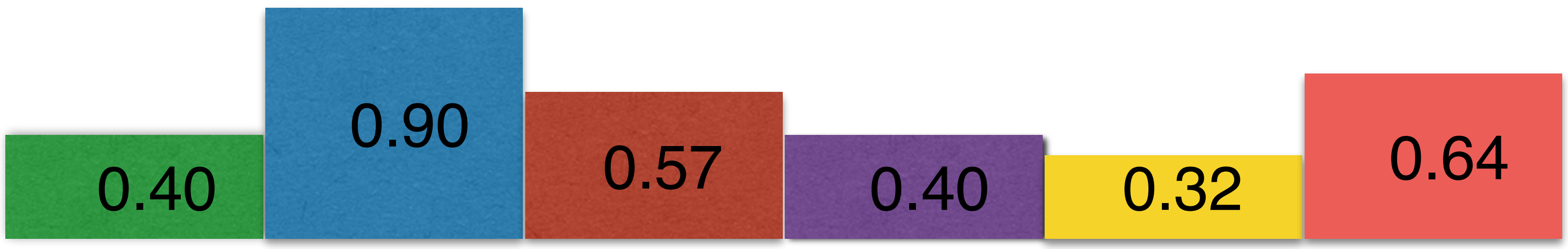


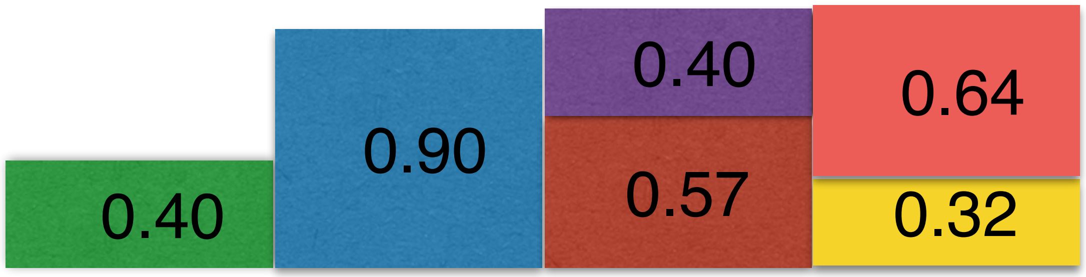
Next Fit Bin Packing





Next Fit Bin Packing





What do you think of this algorithm?

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
- Any two consecutive bins contain items of total size at least one.
- Next Fit Bin Packing returns optimal solutions to bin packing.
- Next Fit Bin Packing runs in polynomial time.
- Next Fit Bin Packing has an approximation ratio of two.

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
  - True! If the total was less, the bin would not have bin split.
- Any two consecutive bins contain items of total size at least one.
- Next Fit Bin Packing returns optimal solutions to bin packing.
- Next Fit Bin Packing runs in polynomial time.
- Next Fit Bin Packing has an approximation ratio of two.

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
  - True! If the total was less, the bin would not have bin split.
- Any two consecutive bins contain items of total size at least one.
  - True! This was true when the second bin was created, and we never removed any items.
- Next Fit Bin Packing returns optimal solutions to bin packing.
- Next Fit Bin Packing runs in polynomial time.
- Next Fit Bin Packing has an approximation ratio of two.

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
  - True! If the total was less, the bin would not have bin split.
- Any two consecutive bins contain items of total size at least one.
  - True! This was true when the second bin was created, and we never removed any items.
- Next Fit Bin Packing returns optimal solutions to bin packing.
  - False! We already saw it return a sub-optimal solution in the first example.
- Next Fit Bin Packing runs in polynomial time.
- Next Fit Bin Packing has an approximation ratio of two.

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
  - True! If the total was less, the bin would not have bin split.
- Any two consecutive bins contain items of total size at least one.
  - True! This was true when the second bin was created, and we never removed any items.
- Next Fit Bin Packing returns optimal solutions to bin packing.
  - False! We already saw it return a sub-optimal solution in the first example.
- Next Fit Bin Packing runs in polynomial time.
  - True! Only the current bin needs to be checked, so it runs in linear time.
- Next Fit Bin Packing has an approximation ratio of two.

Top Hat

Which of the following are true about the Next Fit Bin Packing Algorithm?

- After starting a new current bin, the current and previous bin together contain items of at least size 1.
  - True! If the total was less, the bin would not have bin split.
- Any two consecutive bins contain items of total size at least one.
  - True! This was true when the second bin was created, and we never removed any items.
- Next Fit Bin Packing returns optimal solutions to bin packing.
  - False! We already saw it return a sub-optimal solution in the first example.
- Next Fit Bin Packing runs in polynomial time.
  - True! Only the current bin needs to be checked, so it runs in linear time.
- Next Fit Bin Packing has an approximation ratio of two.
  - True! We have most of the argument already.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

If $NF(L) = 2$, then $OPT(L) = 2$.

- First bin split implies $\sum s_i > 1$ so $OPT(L) \geq 2$.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

If $NF(L) = 2$, then $OPT(L) = 2$.

- First bin split implies $\sum s_i > 1$ so $OPT(L) \geq 2$.

If $NF(L) = 3$, then $OPT(L) \geq 2$.

- Same lower bound, but looser upper bound on $OPT(L)$.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

If $NF(L) = 2$, then $OPT(L) = 2$.

- First bin split implies $\sum s_i > 1$ so $OPT(L) \geq 2$.

If $NF(L) = 3$, then $OPT(L) \geq 2$.

- Same lower bound, but looser upper bound on $OPT(L)$.

If $NF(L) = 2k > 0$, then $OPT(L) \geq k + 1$.

- Because k non-overlapping pairs adding up to more than one.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

If $NF(L) = 2$, then $OPT(L) = 2$.

- First bin split implies $\sum s_i > 1$ so $OPT(L) \geq 2$.

If $NF(L) = 3$, then $OPT(L) \geq 2$.

- Same lower bound, but looser upper bound on $OPT(L)$.

If $NF(L) = 2k > 0$, then $OPT(L) \geq k + 1$.

- Because k non-overlapping pairs adding up to more than one.

If $NF(L) = 2k + 1$, then $OPT(L) \geq k + 1$.

- Because k non-overlapping pairs adding up to more than one plus one more.

Next Fit Bin Packing - Approximation Ratio

If $NF(L) = 1$, then $OPT(L) = 1$.

- Everything fit in the first bin.

If $NF(L) = 2$, then $OPT(L) = 2$.

- First bin split implies $\sum s_i > 1$ so $OPT(L) \geq 2$.

If $NF(L) = 3$, then $OPT(L) \geq 2$.

- Same lower bound, but looser upper bound on $OPT(L)$.

If $NF(L) = 2k > 0$, then $OPT(L) \geq k + 1$.

- Because k non-overlapping pairs adding up to more than one.

If $NF(L) = 2k + 1$, then $OPT(L) \geq k + 1$.

- Because k non-overlapping pairs adding up to more than one plus one more.

So, $NF(L) \leq 2\ OPT(L) - 1$

Bin Packing - Is a Better Approximation Possible?

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2} \sum s_i$?

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2}\sum s_i$?
- $PARTITION \leq_P BINPACKING$

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2}\sum s_i$?
- $PARTITION \leq_P BINPACKING$
- Reduce to bin packing of $S' = \left\{\frac{s_1}{T}, \ldots, \frac{s_n}{T}\right\}$ where $T = \frac{1}{2}\sum s_i$

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2}\sum s_i$?
- $PARTITION \leq_P BINPACKING$
- Reduce to bin packing of $S' = \left\{\frac{s_1}{T}, \ldots, \frac{s_n}{T}\right\}$ where $T = \frac{1}{2}\sum s_i$
- Optimal bin packing answer is 2 if there is a partitioning and 3 otherwise.

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2}\sum s_i$?
- $PARTITION \leq_P BINPACKING$
- Reduce to bin packing of $S' = \left\{ \frac{s_1}{T}, \ldots, \frac{s_n}{T} \right\}$ where $T = \frac{1}{2}\sum s_i$
- Optimal bin packing answer is 2 if there is a partitioning and 3 otherwise.
- If $\alpha < 1.5$, then we can distinguish these cases.

Bin Packing - Is a Better Approximation Possible?

Suppose we had a polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$.

How fast could we solve partitioning?

- Given a multiset of integers $S = \{s_1, \ldots, s_n\}$, is there a subset of them adding up to $\frac{1}{2} \sum s_i$?
- $PARTITION \leq_P BINPACKING$
- Reduce to bin packing of $S' = \left\{ \frac{s_1}{T}, \ldots, \frac{s_n}{T} \right\}$ where $T = \frac{1}{2} \sum s_i$
- Optimal bin packing answer is 2 if there is a partitioning and 3 otherwise.
- If $\alpha < 1.5$, then we can distinguish these cases.
- But we also said that partitioning is NP-Complete.
  - So that would mean $P = NP$.

Bin Packing - Is a Better Approximation Possible?

A polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$ would be a major breakthrough.

- Any chance for $1.5 \leq \alpha < 2$?

Bin Packing - Is a Better Approximation Possible?

A polynomial time bin packing algorithm with an approximation ratio $\alpha < 1.5$ would be a major breakthrough.

- Any chance for $1.5 \leq \alpha < 2$?

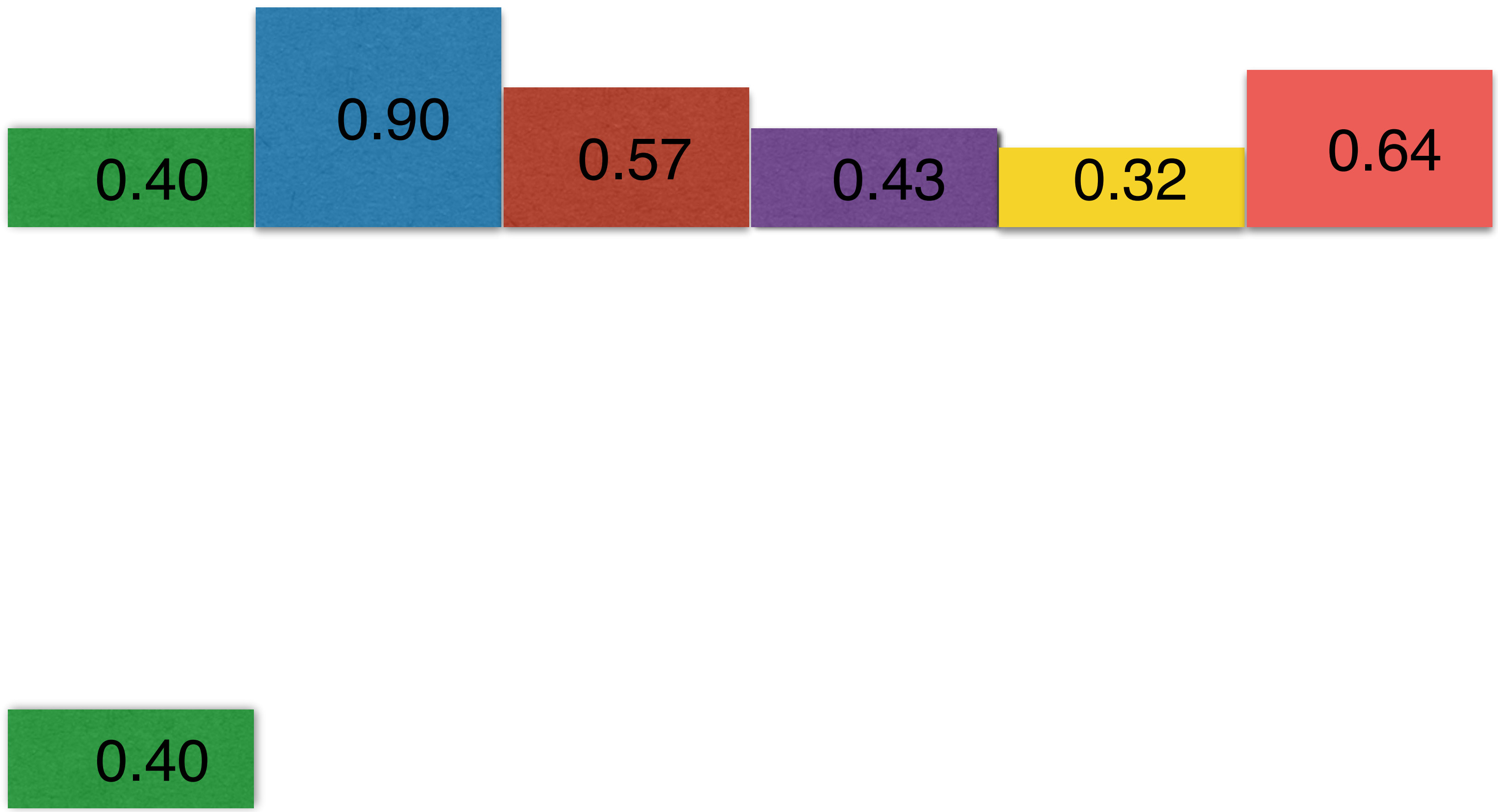
- Two simple algorithms First Fit and Best Fit have $\alpha = 1.7$.

First Fit Bin Packing

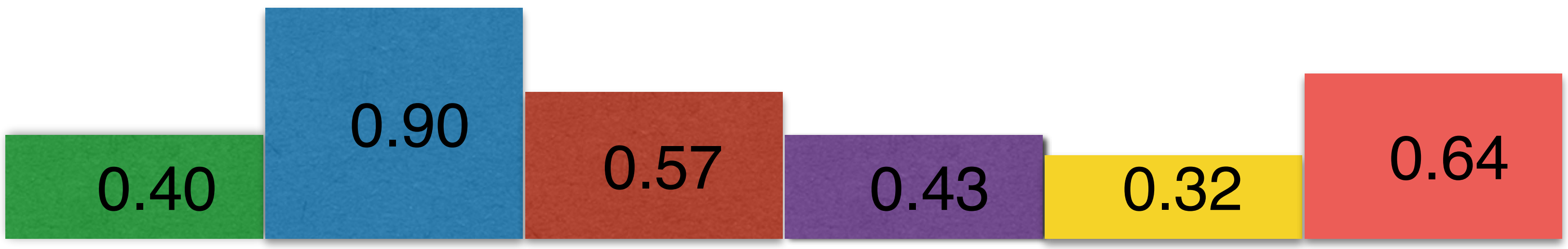
```
Initialize bin list to an empty list.
For each item:
    If any bin has room for the item,
       Put the item in the first bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

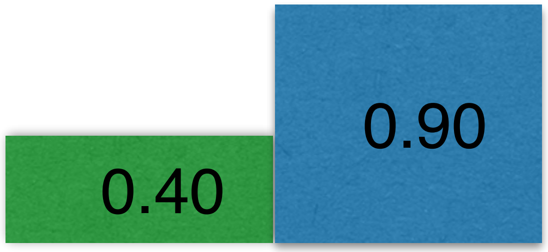
This can be implemented in $O(n \log n)$ time.

First Fit Bin Packing

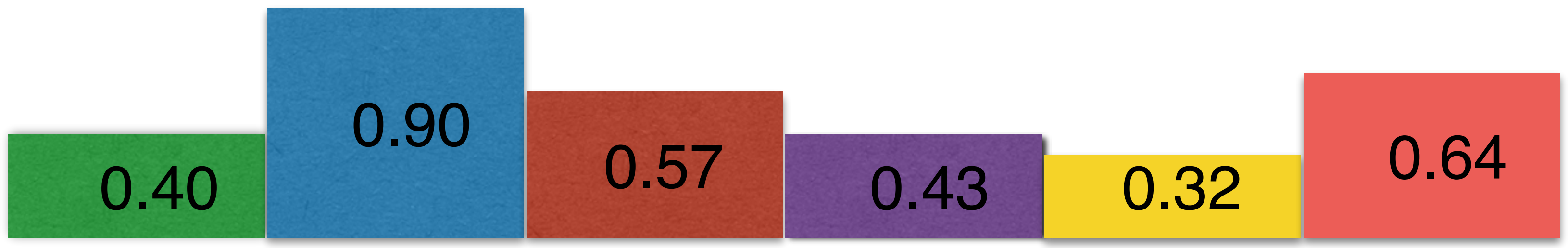


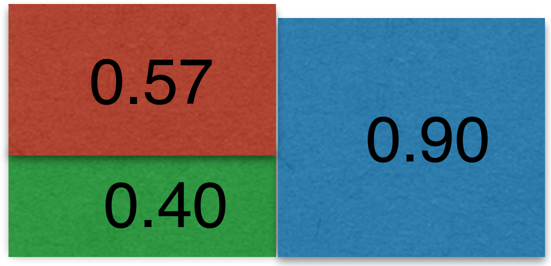
First Fit Bin Packing



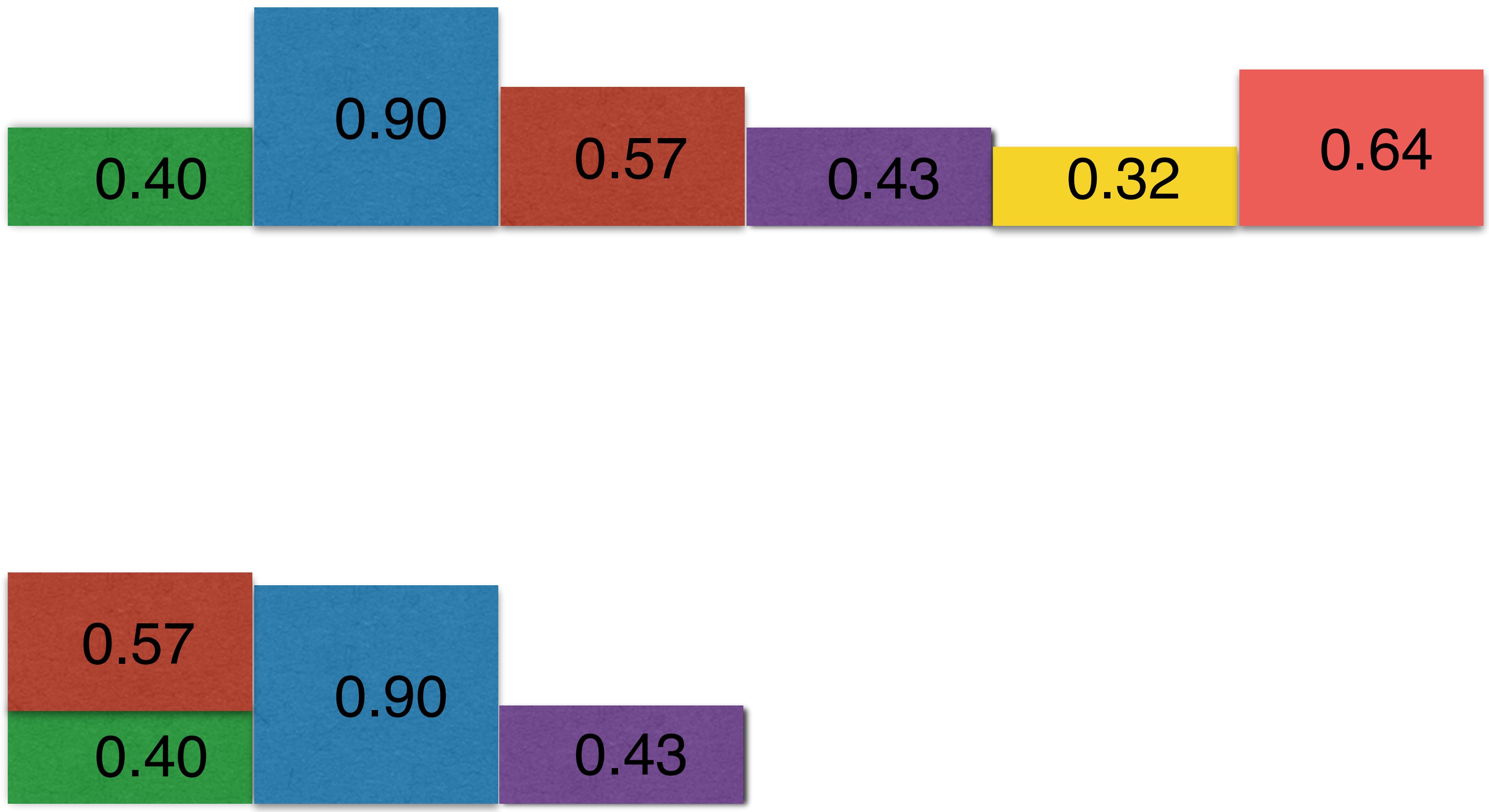


First Fit Bin Packing

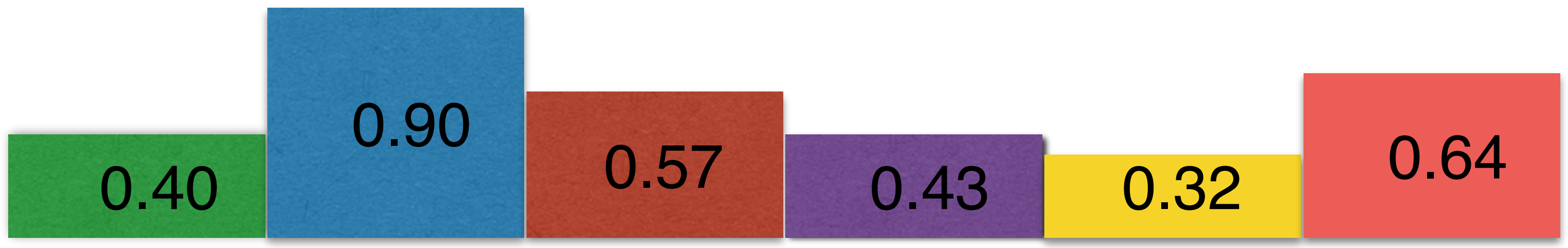




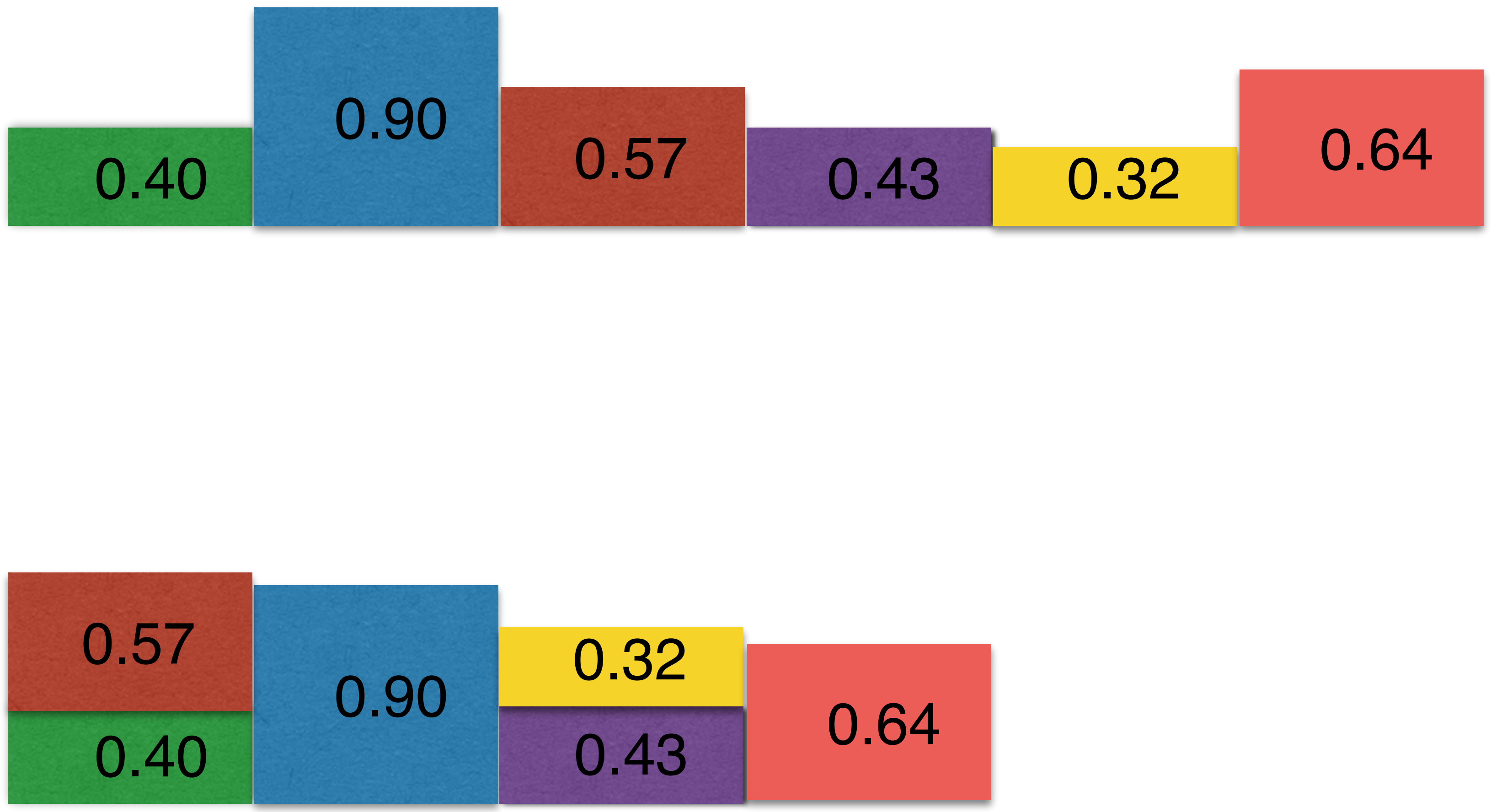
First Fit Bin Packing



First Fit Bin Packing



First Fit Bin Packing



First Fit Bin Packing - Approximation Ratio

- 1971 - $FF(S) \leq 1.7OPT(S) + 3$
- 1972 - $FF(S) \leq 1.7OPT(S) + 2$
- 1976 - $FF(S) \leq 1.7OPT(S) + 0.9$
- 2010 - $FF(S) \leq 1.7OPT(S) + 0.7$
- 2013 - $FF(S) \leq \lfloor 1.7OPT(S) \rfloor$

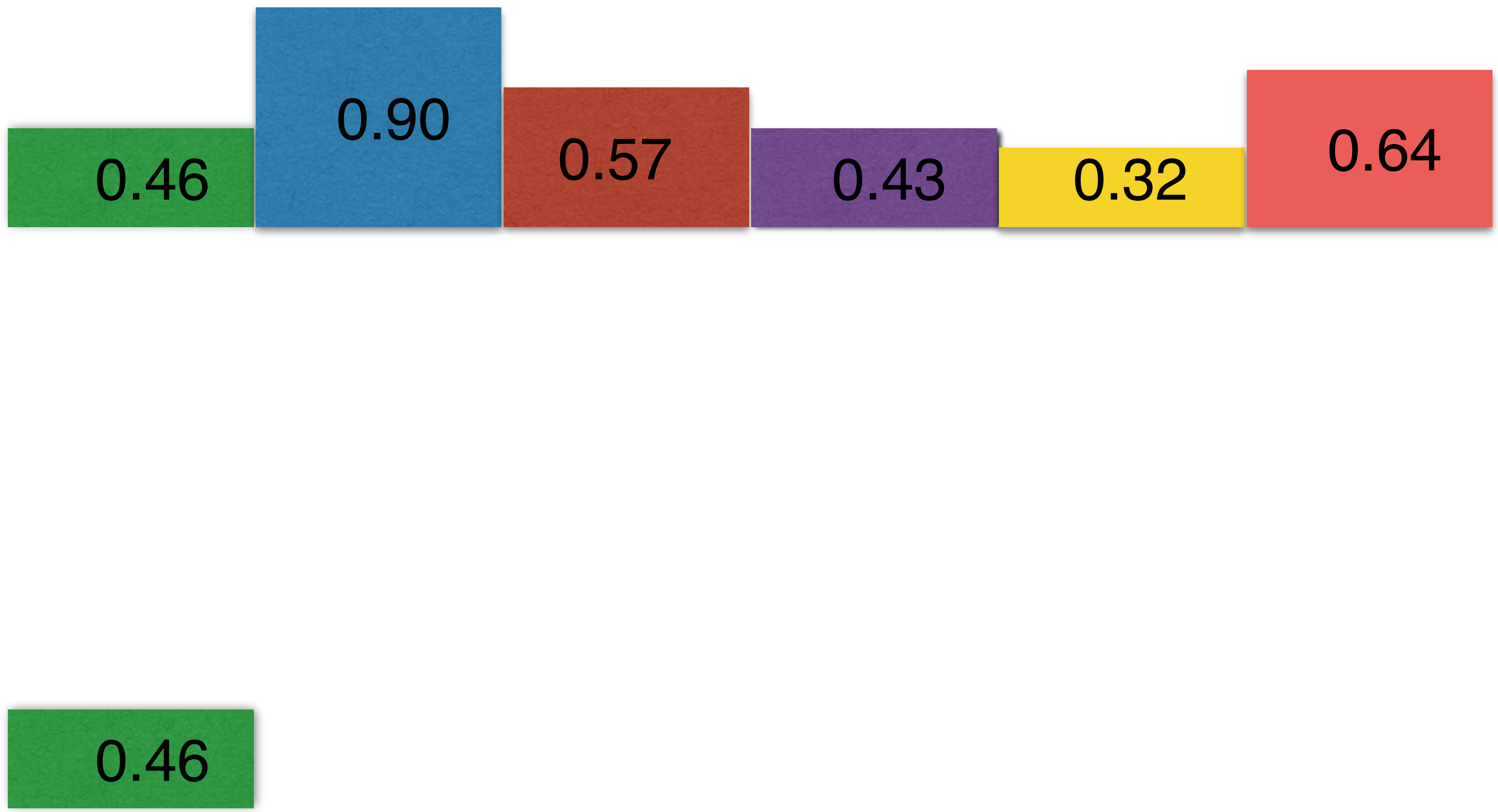
Last one is tight.

Best Fit Bin Packing

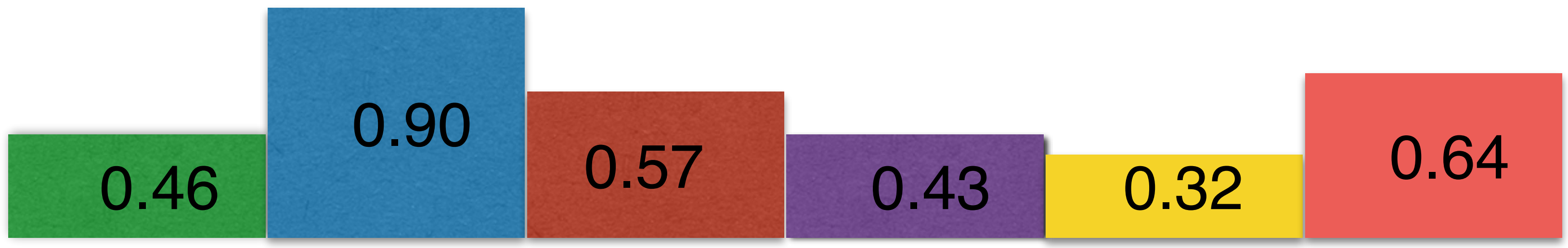
```
Initialize bin list to an empty list.
For each item:
    If any bin has room for the item,
       Put the item in the most heavily loaded bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

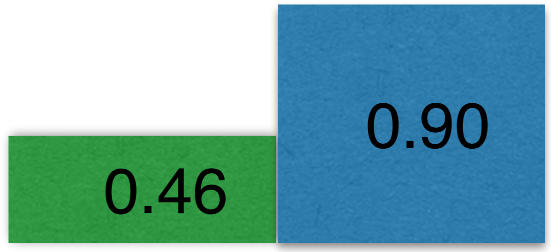
This can be implemented in $O(n \log n)$ time.

Best Fit Bin Packing

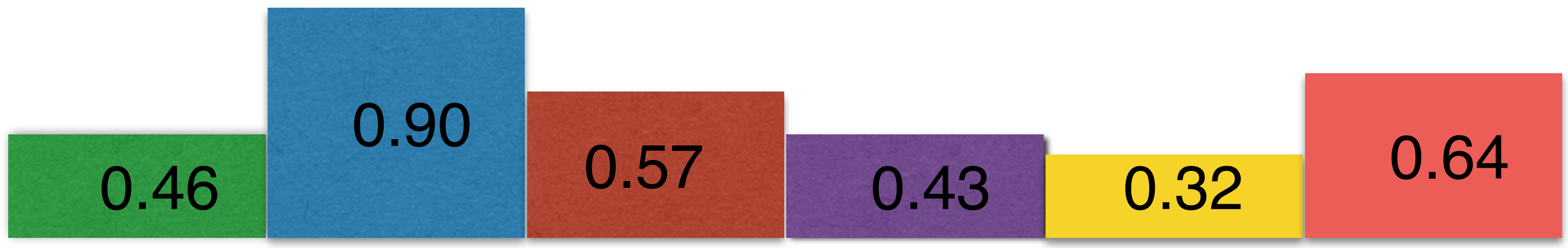


Best Fit Bin Packing

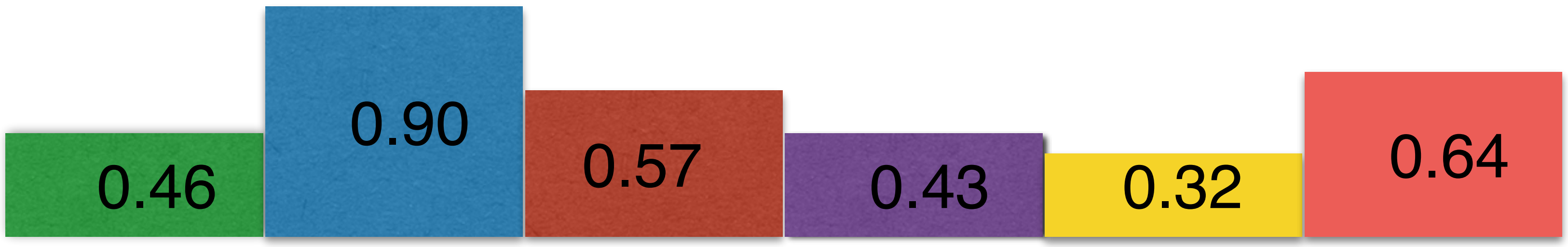




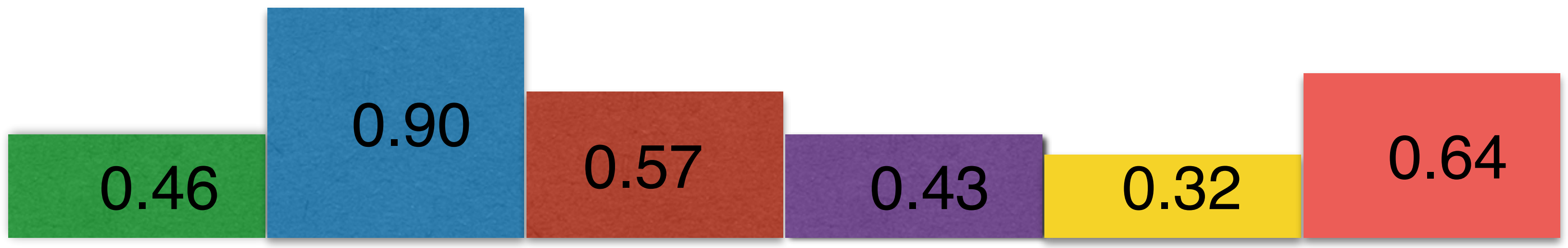
Best Fit Bin Packing



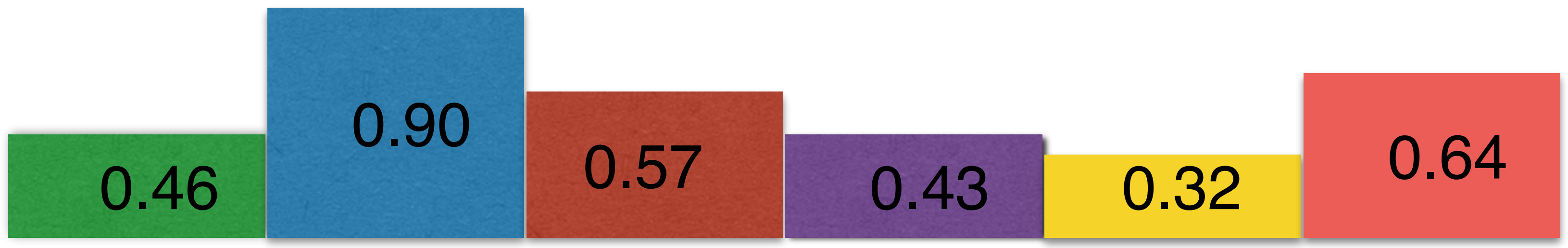
Best Fit Bin Packing

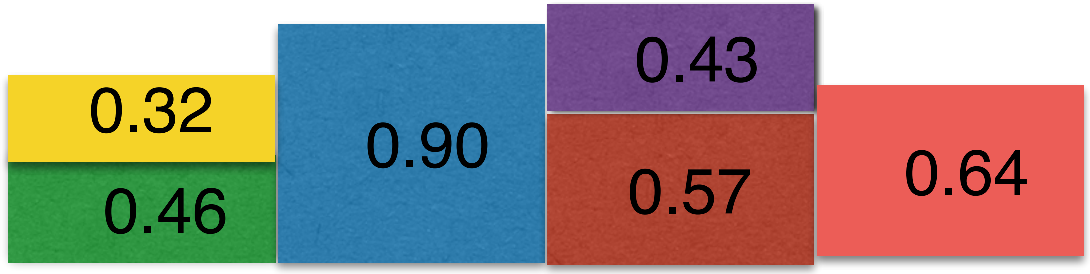


Best Fit Bin Packing



Best Fit Bin Packing





Best Fit Bin Packing

```
Initialize bin list to an empty list.
For each item:
    If any bin has room for the item,
       Put the item in the most heavily loaded bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

This can be implemented in $O(n \log n)$ time.

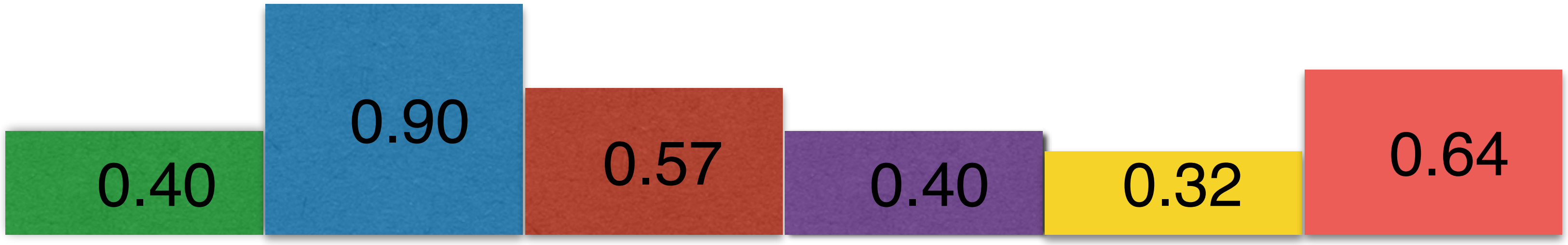
Similar 1971 to 2013 history of approximation analysis to get same $\alpha = 1.7$.

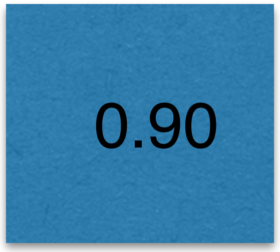
First Fit Decreasing Bin Packing

```
Initialize bin list to an empty list.
Sort items in decreasing order by size.
For each item:
    If any bin has room for the item,
       Put the item in the first bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

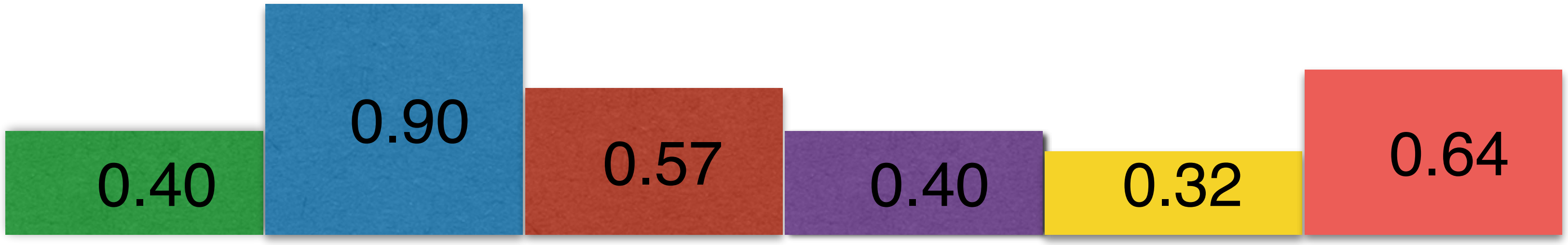
This can be implemented in $O(n \log n)$ time.

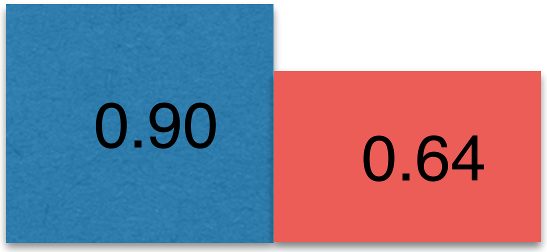
First Fit Decreasing Bin Packing



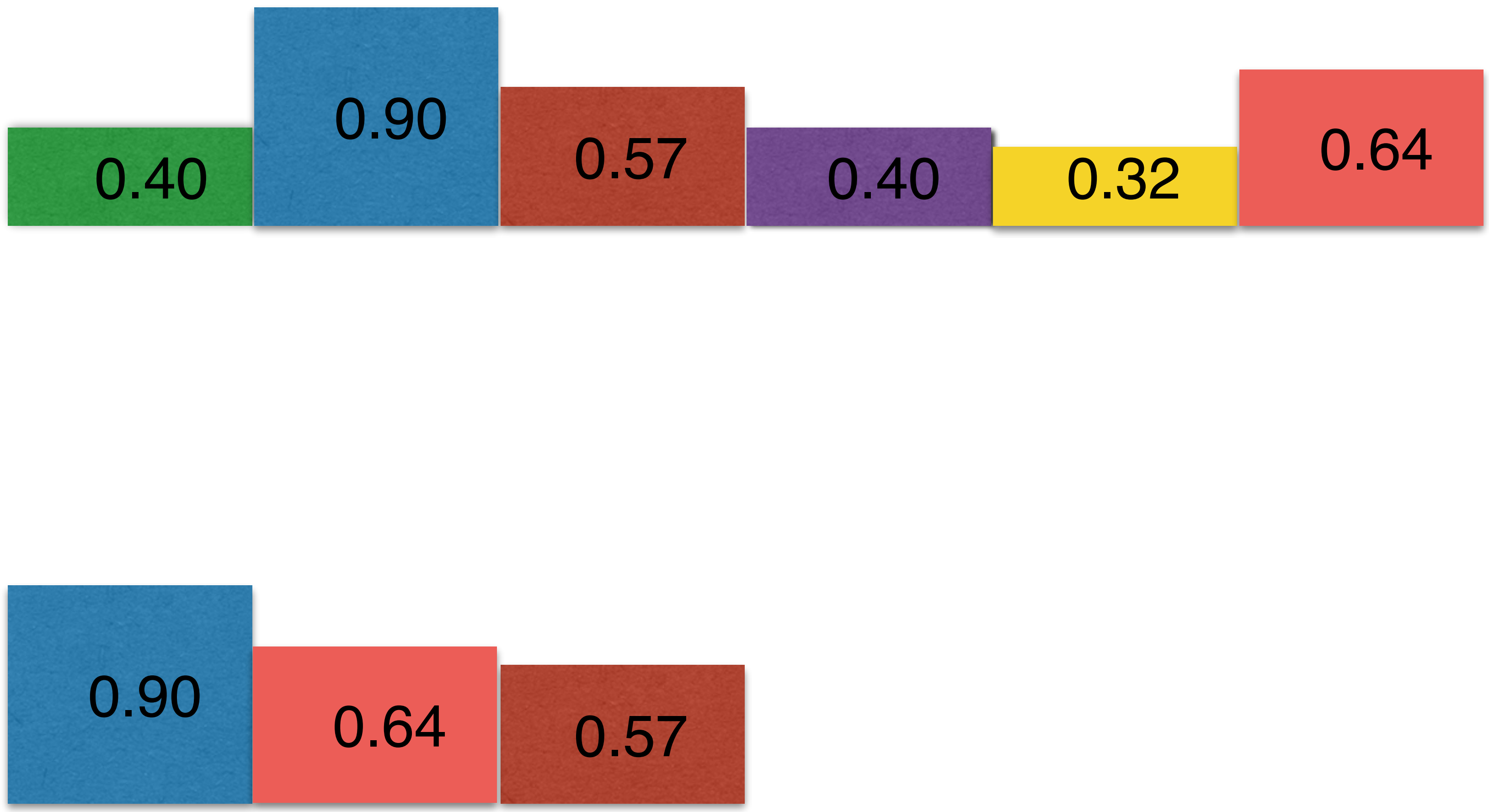


First Fit Decreasing Bin Packing

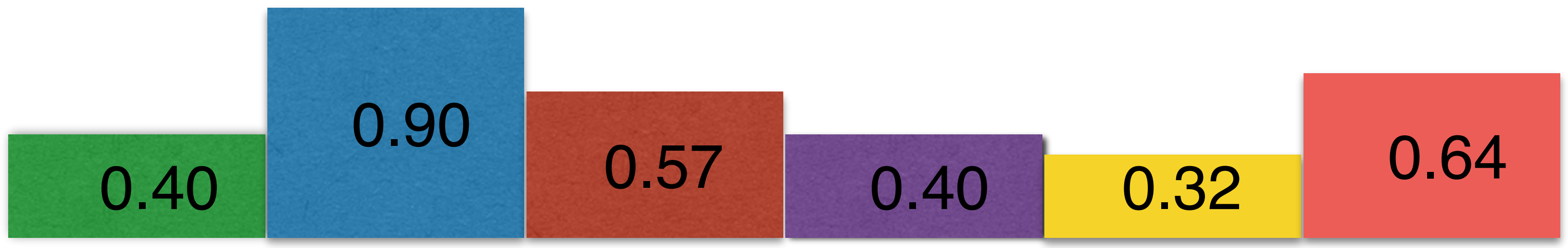


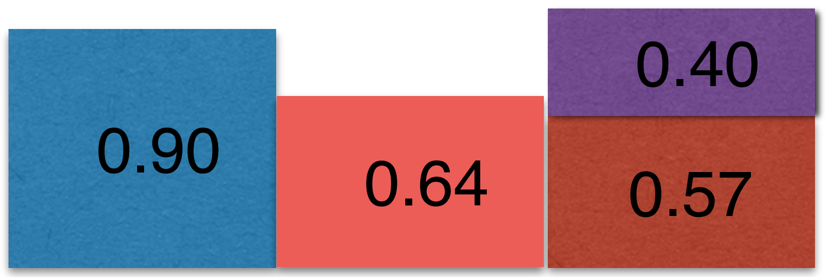


First Fit Decreasing Bin Packing

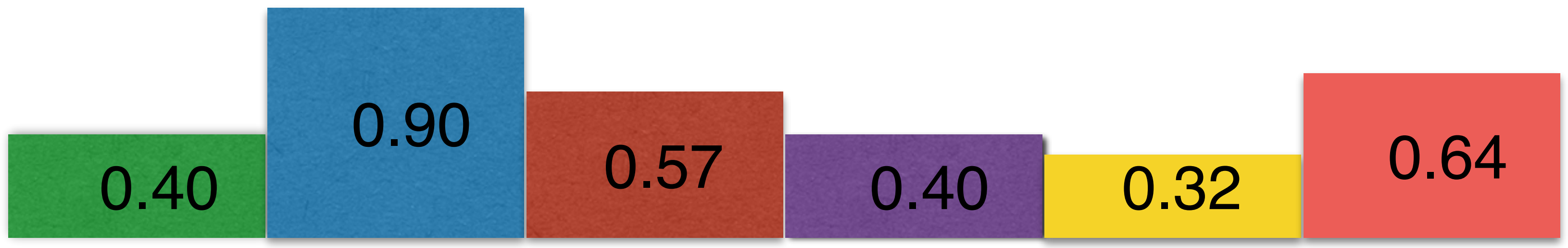


First Fit Decreasing Bin Packing

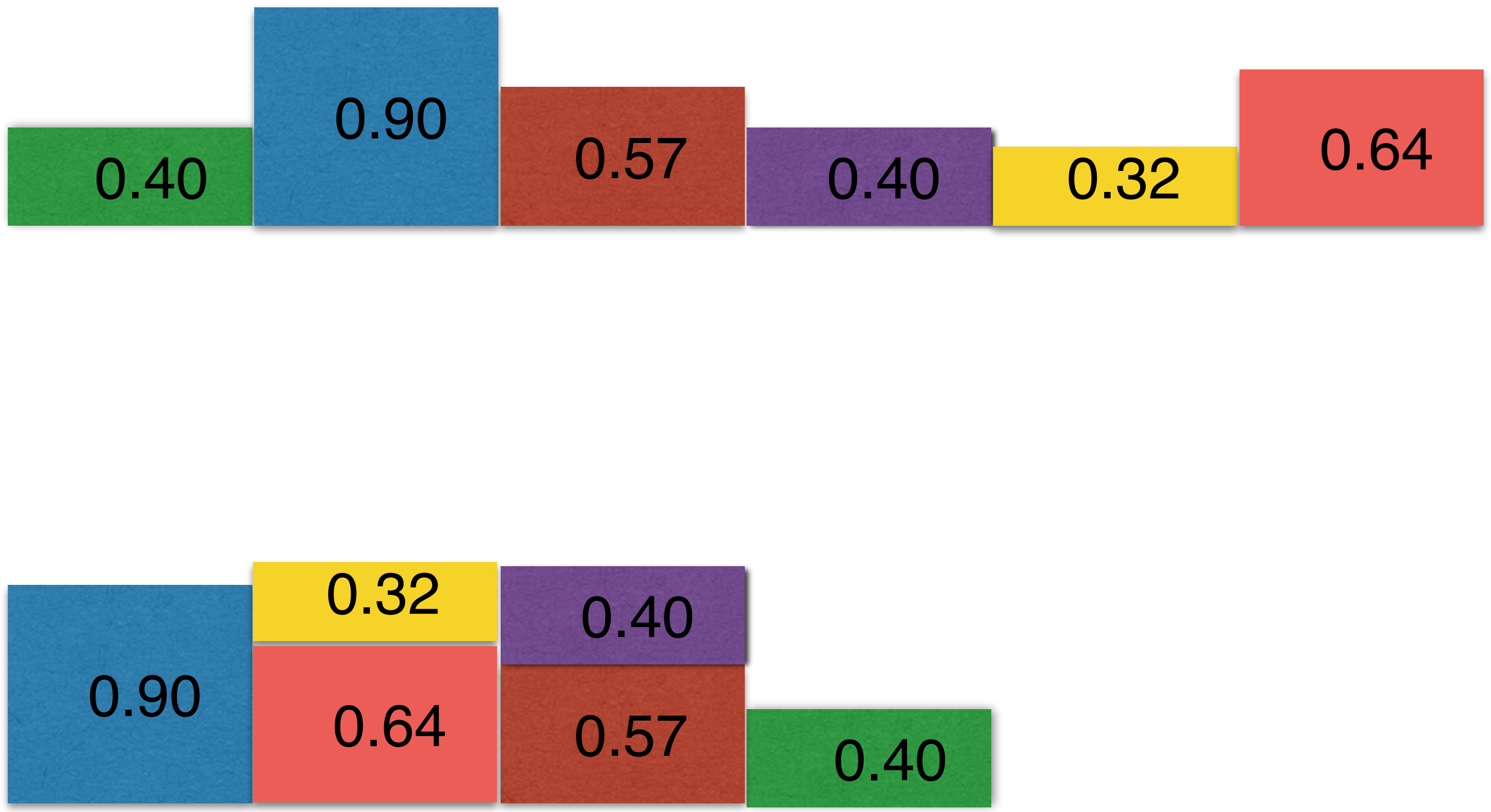




First Fit Decreasing Bin Packing



First Fit Decreasing Bin Packing



First Fit Decreasing Bin Packing

```
Initialize bin list to an empty list.
Sort items in decreasing order by size.
For each item:
    If any bin has room for the item,
       Put the item in the first bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

This can be implemented in $O(n \log n)$ time.

$$FFD(L) \leq \frac{11}{9} OPT(L) + \frac{6}{9}$$

First Fit Decreasing Bin Packing

```
Initialize bin list to an empty list.
Sort items in decreasing order by size.
For each item:
    If any bin has room for the item,
       Put the item in the first bin with room.
    Otherwise,
        Start a new bin.
        Add the item to the new bin.
```

This can be implemented in $O(n \log n)$ time.

$$FFD(L) \leq \frac{11}{9} OPT(L) + \frac{6}{9}$$

If $OPT(L) = 2$, this only gives $FFD(L) \leq 3$, so no trivial $P = NP$.