- deadlock recovery
- deadlock prevention
- deadlock avoidance
- race conditions
- starvation
Dhairght
- Cushearnan speak after Government is
- Cushearnan began up to a limit
- End in money
- bouche a chip
\[ \begin{align*}
\Delta & = \left[ \begin{array}{cc} 1 & 0 \\ 0 & \delta \end{array} \right] \\
\Delta & \rightarrow \left[ \begin{array}{cc} 1 \delta \end{array} \right] \\
\end{align*} \]
The document contains handwritten mathematical expressions. The expressions are not clearly visible due to the handwriting style and the use of symbols and Greek letters. Without clear handwriting, it is challenging to transcribe the exact mathematical content accurately.
Do not go there.
\[ (n - \text{cost}) \]

\[ \text{current need} \]

\[ \text{Need} \]

\[ \text{Allocation} \]

\[ \text{Array} - \text{max} \] 

\[ \text{A} \]

\[ \text{Allocated} \]

\[ \text{int} \]

\[ \text{Head} \]
def recur[i] = recur[i+1], \text{if } x = \text{recur}[i] - x

The recur \text{ to work} \rightarrow x - x\text{ and recur to if } \text{ recur} \rightarrow \text{work} \rightarrow \text{and recur to}

\text{temporarily}

\text{else return safely otherwise}

\text{if } x \geq \text{ recur } \rightarrow \text{def}

\text{Customer is worthy if } x
I can't come back.

First, I'm yellow.

Take it.

$f(x) = t$

For the sake of work,

need $[?] \rightarrow \text{work}$

$[?] \rightarrow \text{false}$

Find such that

I found someone!!

Money back!
\[ \text{Complexity } o(n) \]
algo

\rightarrow \text{SAFE} \rightarrow \text{allocate} \rightarrow \text{avail} = \text{avail} - x

\text{or}

\text{already done} \begin{cases} \text{alloc}[i] = \text{alloc}[i] + x \\ \text{need}[i] = \text{need}[i] = x \end{cases}

\text{done} \rightarrow \text{return 'OK'}

\text{unsafe} \rightarrow
A default list

def a dummy parameter (x, y)
def a dummy parameter (x, y)

\[ \text{new} \times = \text{new} \times \times + \text{alloc} \times \times - \text{alloc} \times \times \]
process i returns $x$

$\rightarrow$ avail = avail + x

alloc[i] = alloc[i] - x;

need[i] = 0

\(O(n^3)\)

look at each entry in deferred list

$\rightarrow$ run safety algo

$\rightarrow$ if safe allocate & delete from list

$\rightarrow$ next item
a web (wont be greedy)

The bag can be simplified to

- every resource has one instance

Special case

Deduced by default
Could be decreased

no necessary & soft

Check in WEG