

```

(par
  ((i_1 ()))
  (seq
    (choose
      k
      unit
      (backtrack i_2 k_2)
      (err "Failure!")))
  (let x =
    (choose
      k_1
      1
      (seq
        (backtrack i_2 k_3)
        4)
      (backtrack i_1 k))
    in
    (let y =
      (recv i_2)
      in
      (let z =
        (recv i_3)
        in
        (if (< x y)
          (if (< y z)
            (seq
              (collect k)
              (collect k_1)
              (send i_2 1)
              (send i_3 1)
              (err
                "Success!")))
            (backtrack
              i_1
              1)))

```

```

(par
  ((i_1 ()))
  (seq
    (choose
      k
      unit
      (backtrack i_2 k_2)
      (err "Failure!")))
  (let x =
    (choose
      k_1
      1
      (seq
        (backtrack i_2 k_3)
        4)
      (backtrack i_1 k))
    in
    (let y =
      (recv i_2)
      in
      (let z =
        (recv i_3)
        in
        (if (< x y)
          (if (< y z)
            (seq
              (collect k)
              (collect k_1)
              (send i_2 1)
              (send i_3 1)
              (err
                "Success!")))
            (backtrack
              i_1
              1)))

```

```

(par
  ((i_1 ()))
  (seq
    (choose
      k
      unit
      (backtrack i_2 k_2)
      (err "Failure!")))
  (let x =
    (choose
      k_1
      1
      (seq
        (backtrack i_2 k_3)
        4)
      (backtrack i_1 k))
    in
    (let y =
      (recv i_2)
      in
      (let z =
        (recv i_3)
        in
        (if (< x y)
          (if (< y z)
            (seq
              (collect k)
              (collect k_1)
              (send i_2 1)
              (send i_3 1)
              (err
                "Success!")))
            (backtrack
              i_1
              1)))

```

se

track i_2 k_2)
Failure!"))

=
use

cktrack i_2 k_3)

ktrack i_1 k))

=
v i_2)

z =
recv i_3)

(< x y)
f (< y z)
(seq
 (collect k)
 (collect k_1)
 (send i_2 1)
 (send i_3 1)
 (err
 "Success!"))
(backtrack
 i_1
 1))

```
(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!"))
        unit)
      (let x =
        (choose
          k_1
          1
          (seq
            (backtrack i_2 k_3)
            4)
            (backtrack i_1 k))
        in
        (let y =
          (recv i_2)
          in
          (let z =
            (recv i_3)
            in
            (if (< x y)
              (if (< y z)
                (seq
                  (collect k)
                  (collect k_1)
                  (send i_2 1)
                  (send i_3 1)
                  (err
                    "Success!"))
                (backtrack
                  i_1
                  1
```

```
(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!"))
        unit)
      (let x =
        (choose
          k_1
          1
          (seq
            (backtrack i_2 k_
            4)
            (backtrack i_1 k))
        in
        (let y =
          (recv i_2)
          in
          (let z =
            (recv i_3)
            in
            (if (< x y)
              (if (< y z)
                (seq
                  (collect k)
                  (collect k_1)
                  (send i_2 1)
                  (send i_3 1)
                  (err
                    "Success!"))
                (backtrack
                  i_1
                  1
```

k_2)

_2 k_3)

1 k))

k)

k_1)

2 1)

3 1)

ss!"))

ck

```
(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!")
          unit)
        (let x =
          (choose
            k_1
            1
            (seq
              (backtrack i_2 k_3)
              4)
            (backtrack i_1 k))
          in
            (let y =
              (recv i_2)
              in
                (let z =
                  (recv i_3)
                  in
                    (if (< x y)
                      (if (< y z)
                        (seq
                          (collect k)
                          (collect k_1)
                          (send i_2 1)
                          (send i_3 1)
                          (err
                            "Success!"))
                        (backtrack
                          i_1
```

```
(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!")
          unit)
        (let x =
          (choose
            k_1
            1
            (seq
              (backtrack i_2 k_3)
              4)
            (backtrack i_1 k))
          in
            (let y =
              (recv i_2)
              in
                (let z =
                  (recv i_3)
                  in
                    (if (< x y)
                      (if (< y z)
                        (seq
                          (collect k)
                          (collect k_1)
                          (send i_2 1)
                          (send i_3 1)
                          (err
                            "Success!"))
                        (backtrack
                          i_1
```

```
(pa
  ((i_
    ((
```

```

(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!")
          unit)
        (let x =
          (choose
            k_1
            1
            (seq
              (backtrack i_2 k_3)
              4)
            (backtrack i_1 k))
          in
            (let y =
              (recv i_2)
              in
                (let z =
                  (recv i_3)
                  in
                    (if (< x y)
                      (if (< y z)
                        (seq
                          (collect k)
                          (collect k_1)
                          (send i_2 1)
                          (send i_3 1)
                          (err
                            "Success!"))
                        (backtrack
                          i_1

```

```

(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!")
          unit)
        (let x =
          (choose
            k_1
            1
            (seq
              (backtrack i_2 k_3)
              4)
            (backtrack i_1 k))
          in
            (let y =
              (recv i_2)
              in
                (let z =
                  (recv i_3)
                  in
                    (if (< x y)
                      (if (< y z)
                        (seq
                          (collect k)
                          (collect k_1)
                          (send i_2 1)
                          (send i_3 1)
                          (err
                            "Success!"))
                        (backtrack
                          i_1

```

```

(par
  ((i_1
    ((k
      (seq
        (choose
          k
          (backtrack i_2 k_2)
          (err "Failure!")
          unit)
        (let x =
          (choose
            k_1
            1
            (seq
              (backtrack i_2 k_3)
              4)
            (backtrack i_1 k))
          in
            (let y =
              (recv i_2)
              in
                (let z =
                  (recv i_3)
                  in
                    (if (< x y)
                      (if (< y z)
                        (seq
                          (collect k)
                          (collect k_1)
                          (send i_2 1)
                          (send i_3 1)
                          (err
                            "Success!"))
                        (backtrack
                          i_1

```


