

Making fast faster

Low-level optimisation

Bruce Merry

-
-
-

General advice

General advice

- Don't do it.

General advice

- Don't do it.
- (For advanced programmers) Don't do it yet.

General advice

- Don't do it.
- (For advanced programmers) Don't do it yet.
- Keep your old version around to compare to.

General advice

- Don't do it.
- (For advanced programmers) Don't do it yet.
- Keep your old version around to compare to.
- Only optimise the inner loops.

Quick and dirty

- Use 32-bit integers (Pascal: `longint`).
- Arrays
 - Smaller elements improve speed (don't overflow!).
 - Walk along rows of 2D arrays, not columns.
- Use `memset`/`fillchar`.
- C/C++: use `inline` on tiny functions.

Sentinels

Sentinels are special values you use to eliminate checks in a loop.

Sentinels

Sentinels are special values you use to eliminate checks in a loop.

- If searching for a 0 in an array, place one at the end.

Sentinels

Sentinels are special values you use to eliminate checks in a loop.

- If searching for a 0 in an array, place one at the end.
- If searching for the smallest valid array member, make sure the invalid ones are all huge.

Sentinels

Sentinels are special values you use to eliminate checks in a loop.

- If searching for a 0 in an array, place one at the end.
- If searching for the smallest valid array member, make sure the invalid ones are all huge.
- In a 2D maze problem, place walls all around the outside.