Item 201
git_comments:

1. Check if string tail is full ASCII (common case, fast)
2. size == 0
3. Fall back to UTF8 validation of tail string.

git_commits:

1. **summary:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   message: ARROW-10313: [C++] Faster UTF8 validation for small strings This improves CSV string conversion performance by about 30%. Closes #8470 from pitrou/ARROW-10313-faster-utf8-validate Authored-by: Antoine Pitrou <antoine@python.org> Signed-off-by: Antoine Pitrou <antoine@python.org>
   label: code-design

github_issues:

github_issues_comments:

github_pulls:

1. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
2. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
   label: code-design
3. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
   label: code-design
4. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
5. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
6. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.
7. **title:** ARROW-10313: [C++] Faster UTF8 validation for small strings
   body: This improves CSV string conversion performance by about 30%.

github_pulls_comments:

1. https://issues.apache.org/jira/browse/ARROW-10313
2. This also improves the ARROW-10308 benchmark by about 9%.
3. **body:** Another possibility would be to compute and store ASCIIness of values while parsing CSV (ideally this should not cost anything CPU-wise). Then we can reuse that information to skip UTF8 validation for most values. Edit: actually, a quick attempt shows a significant decrease in CSV parsing speed. Too bad.
   label: code-design

github_pulls_reviews:

1. ```suggestion uint32_t head_mask = internal::SafeLoadAs<uint32_t>(data); uint32_t tail_mask = internal::SafeLoadAs<uint32_t>(data + size - 4); if (ARROW_PREDICT_TRUE(((head_mask | tail_mask) & high_bits_32) == 0)) { return true; } ```
2. ```suggestion uint16_t tail_mask = SafeLoadAs<uint16_t>(data + size - 2); uint16_t head_mask = SafeLoadAs<uint16_t>(data); ```
3. ```suggestion uint64_t mask64 = SafeLoadAs<uint64_t>(data); ```

jira_issues:

1. **summary:** [C++] Improve UTF8 validation speed and CSV string conversion
   description: Based on profiling from ARROW-10308, UTF8 validation is a bottleneck of CSV string conversion. This is because we must validate many small UTF8 strings individually.
2. **summary:** [C++] Improve UTF8 validation speed and CSV string conversion
   description: Based on profiling from ARROW-10308, UTF8 validation is a bottleneck of CSV string conversion. This is because we must validate many small UTF8 strings individually.
3. **summary:** [C++] Improve UTF8 validation speed and CSV string conversion
   description: Based on profiling from ARROW-10308, UTF8 validation is a bottleneck of CSV string conversion. This is because we must validate many small UTF8 strings individually.
   label: code-design

jira_issues_comments:

1. Issue resolved by pull request 8470 [https://github.com/apache/arrow/pull/8470]